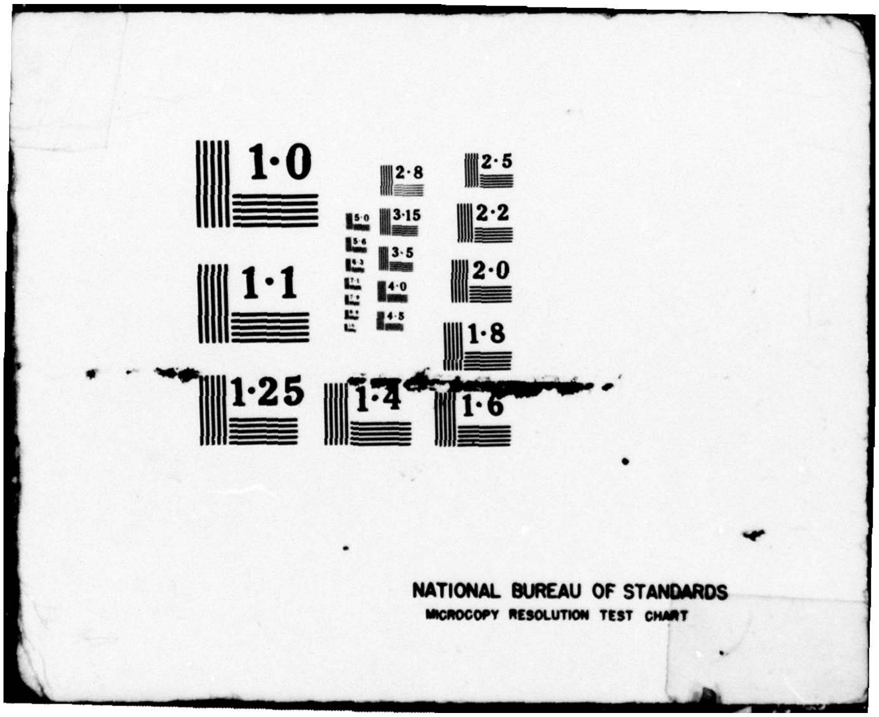
AD-A069 843 LAMONT-DOHERTY GEOLOGICAL OBSERVATORY PALISADES N Y F/G 20/1
MULTIPLE SURFACE SCATTERING FROM THE ICE COVERED ARCTIC OCEAN.(U)
MAR 79 A BEILIS, F D TAPPERT, H W KUTSCHALE N00014-75-C-0595
UNCLASSIFIED LDGO-CU-2-79 NL

| OF |
AD
A069843

END
DATE
FILMED
7-79
DDC


1·0
2·8
2·5
3·15
2·2
5·0
5·6
6·3
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

LAMONT-DOHERTY GEOLOGICAL OBSERVATORY
OF COLUMBIA UNIVERSITY
Palisades, New York 10964

MULTIPLE SURFACE SCATTERING FROM THE ICE COVERED ARCTIC OCEAN

by

Alan Beilis[1]
Department of Physics
New York University
New York, New York 10003

Frederick D. Tappert[2]
Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

Henry W. Kutschale
Lamont-Doherty Geological Observatory
of Columbia University
Palisades, New York 10964

See 1473 in back

March 1979

CU-2-79

Technical Report No. 2

CONTRACT N00014-75-C-0595
supported by
Office of Naval Research

Present Addresses:
[1]Bell Laboratories, Inc., Whippany, N. J. 07981
[2]University of Miami, RSMAS, 4600 Rickenbacker Causeway,Miami,FL 33149

TABLE OF CONTENTS

ABSTRACT

The range dependent boundary condition imposed by a rough ice surface serves to couple the normal modes of an underwater acoustic field. Beginning with the parabolic wave equation, the solution is expanded in terms of normal-mode depth functions modulated by random amplitudes. A system of master equations for the quantities $<a_n(r)a_n^*(r)>$ is derived, where the $a_n(r)$ are the normal-mode amplitudes, and the brackets indicate an ensemble average. The master equations determine the mean power in each mode as a function of range, describing the transfer of energy between modes. Explicit relations for the coupling coefficients are obtained in terms of the spectrum of the rough surface. Transmission loss, intensity redistribution and the approach to equipartition of energy as functions of range and depth are obtained numerically.

I. INTRODUCTION

The problem of acoustic scattering from the rough ice cover of the Arctic Ocean has been of considerable interest for a long time because the effect is unavoidable. To the present, the multiple scattering phenomenon has not been considered with the exception of recent work by Kryazhev,et al[1]. The present work will be compared in detail with the Soviet work in a future report.

The present calculation makes use of the method of parabolic equations (forward scattering, long range, $k_o r >> 1$) introduced to underwater acoustics by F. D. Tappert[2] to study long-range propagation. The medium we consider is the Arctic Ocean whose sound speed profile is taken to be bilinear[3]. The shape of the interface between the under-ice cover and the water is $\eta(r)$, where $\eta(r)$ is a random ice roughness function represented by its wavenumber distribution. The integral of this wavenumber distribution is the r.m.s. under-ice amplitude squared. The form of the wavenumber distribution is motivated by previous research carried out by Hibler, et al[4] and is taken here to be Gaussian.

The rough surface causes the initial acoustic intensity to be redistributed as a function of depth. It is also found that the energy per mode approaches equipartition as the observation range approaches long ranges.

The reader is referred to Ref. 5 to obtain further details to the calculations shown below. It is noted here that this is a preliminary report to be followed by a report including comparison to experimental data.

II. Theory.

Consider a point source at point $(0,z_0)$ in a cylindrical coordinate system, emitting an acoustic signal at angular frequency ω. The problem is to describe the average effect of a randomly distributed rough surface on the distribution of energy within the ocean waveguide. Of particular interest is the average intensity as a function of range and depth $I(r,z) = \langle p(r,z)p^*(r,z)\rangle$ where $p(r,z)$ is the pressure amplitude.

We begin with the equation satisfied by the envelope modulating the cylindrical plane wave, i.e. the parabolic wave equation:

$$2ik_0\partial\psi(r,z)/\partial r + \partial^2\psi(r,z)/\partial z^2 + k_0^2(n^2(z) - 1)\psi(r,z) = 0 \ , \tag{1}$$

where the envelope function $\psi(r,z)$ satisfies the following boundary conditions:

$$\psi(r,\eta(r)) = 0 \ , \tag{1a}$$

$$\partial\psi(r,z)/\partial z\Big|_{z=H} = 0 \ , \tag{1b}$$

$$\psi(0,z) = (8\pi k_0)^{-1/2}\delta(z - z_0) \ , \tag{1c}$$

where $\eta(r)$ represents the randomly distributed boundary between the water and under-ice surface, and H is the ocean depth.

Performing the coordinate transformation[5]

$$r' = r \, ,$$
$$z' = z - \eta(r) \, , \tag{2}$$

and rewriting $\psi(r,z)$ as $\tilde{\psi}(r',z')\exp(i\theta(r',z'))$, Eq. (1) becomes (dropping the primes, suppressing the arguments for convenience and imposing conditions on $\theta(r',z')$)

$$i\partial\tilde{\psi}/\partial r + (1/2k_0)\partial^2\tilde{\psi}/\partial z^2 + (k_0/2)(n^2(z) - 1)\tilde{\psi} - k_0 z\partial^2\eta/\partial r^2\tilde{\psi} = 0 \, , \tag{3}$$

$$\tilde{\psi}(r,0) = 0 \, , \tag{3a}$$

$$\partial\tilde{\psi}/\partial z\Big|_{z=H} = 0 \, , \tag{3b}$$

$$\tilde{\psi}(0,z) = (8\pi k_0)^{-1/2}\delta(z - \bar{z}_0) \, , \tag{3c}$$

where $\bar{z}_0 = z_0 - \eta(0)$. The random boundary condition (Eq. (1a)) becomes a "flat surface" boundary condition at the expense of introducing an additional perturbative term in the parabolic wave equation.

The envelope function, $\tilde{\psi}$, is expanded in terms of a complete set of "states" or normal modes, defined by the unperturbed sound-speed profile. Each normal mode is modulated by a random amplitude indicating the effect of the surface on acoustic propagation:

$$\tilde{\psi}(r,z) = \sum_n a_n(r)\phi_n(z)\exp[i(\lambda_n - k_0/2)r] \ , \qquad (4)$$

where

$$[(1/2k_0)\partial^2/\partial z^2 + (k_0/2)n^2(z)]\phi_n(z) = \lambda_n\phi_n(z) \ ,$$

$$\int_0^H \phi_n(z)\phi_m(z)dz = \delta_{nm} \ .$$

Inserting Eq. (4) into Eq. (3) and using the orthonormality of the normal-mode depth function, one finds, upon equating the coefficients of $\phi_n(z)\exp[i(\lambda_n - k_0/2)r]$, that the random amplitudes satisfy the coupled mode equations,

$$da_n(r)/dr = i\sum_m c_{mn}(r)\exp(i\lambda_{mn}r)a_m(r) \ , \qquad (5)$$

where

$$\lambda_{mn} = \lambda_m - \lambda_n \ ,$$

$$c_{mn}(r) = \int_0^H \phi_m(z)H(r,z)\phi_n(z)dz$$

$$= -k_0 Z_{mn}\partial^2\eta(r)/\partial r^2 \ ,$$

with

$$H(r,z) = -k_0 z\partial^2\eta(r)/\partial r^2 \ ,$$

$$Z_{mn} = Z_{nm} \equiv \int_0^H \phi_m(z)z\phi_n(z)dz \ ,$$

is the "dipole moment". The coupling coefficients, $c_{mn}(r)$, are real and symmetric. The coupling coefficients are expressed in terms of a Fourier integral representation. Imposing the conditions that the mean amplitude of the surface roughness is zero and that correlations of coupling coefficients be spatially stationary, we obtain specific conditions on the Fourier amplitudes. Integrating Eq. (5) from r to r + Δr, where Δr << r, to second order in the coupling coefficients and anticipating use of the above mentioned statistical properties of the Fourier amplitudes, an approximate expression for the $a_n(r)$ is obtained. One then constructs the average correlation amplitude between modes n and m applying the random phase approximation (this quantity appears in the average intensity):

$$\begin{aligned} A_{nm}(r) &= \langle a_n(r) a_m^*(r) \rangle = \langle |a_n(r)|^2 \rangle \delta_{nm} \\ &= A_n(r) \delta_{nm} \ . \end{aligned} \tag{6}$$

Using the expression for $a_n(r)$ obtained by integrating the coupled mode equations, we obtain the master equations governing the evolution of $A_n(r)$ as one steps out in range:

$$dA_n(r)/dr = \sum_{m \neq n} \Gamma^s_{nm} [A_m(r) - A_n(r)] \ , \tag{7}$$

where

$$\Gamma^s_{nm} = \Gamma^s_{mn} = 2\pi k_0^2 |Z_{nm}|^2 \ k^4 P_1(k) \Big|_{k=\lambda_{nm}}$$

with $P_1(k)$ being the under-ice surface wavenumber distribution whose integral over all wavenumbers is the r.m.s. surface height squared.

Eq. (7) is amenable to physical interpretation. If mode n has the largest energy content at some range, then Eq. (7) tells us that as we step out in range, mode n will lose energy to the other modes of the system ($dA_n/dr < 0$). If in the process of releasing this energy, mode $n' \neq n$ obtains a larger share of the energy than the other modes, then a mode n' will release energy to the other modes ($dA_{n'}/dr < 0$). This mixing of energy continues as one marches out in range until sufficient multiple scattering has occurred such that the energy becomes equipartiioned. In that case $A_m(r) \rightarrow A_n(r)$ or $dA_n(r)/dr \underset{r\rightarrow\infty}{\rightarrow} 0$. From Eq. (7), summing over all modes, one finds

$$\frac{d}{dr}\left[\sum_n A_n(r)\right] = 0 \ ,$$

or

$$\sum_n A_n(r) = \text{constant} = \sum_n A_n(0) \quad . \tag{8}$$

In view of Eq. (8) and the interpretation of Eq. (7) we conclude

$$A_n(r) \underset{r\rightarrow\infty}{\rightarrow} \frac{1}{N} \qquad \text{for all } n = 1,N \ , \tag{8a}$$

where N is the total number of modes in the system (restricted to include all propagating modes).

Eq. (8) is a statement of energy conservation, which should be expected since we are considering initially a medium with no loss mechanisms.

III. CALCULATIONS

We begin with the master equations

$$dA_n(r)/dr = \sum_{m \neq n} \Gamma^s_{nm}\big(A_m(r) - A_n(r)\big). \tag{9}$$

Defining

$$\Lambda_{nm} = \big(1 - \delta_{nm}\big)\Gamma^s_{nm} - \delta_{nm} \sum_{\ell \neq n} \Gamma^s_{n\ell} \tag{10}$$

where ℓ is a dummy index , Eq. (9) can be rewritten as

$$dA_n(r)/dr = \sum_m \Lambda_{nm} A_m(r) \tag{11}$$

or in matrix notation

$$dA(r)/dr = \Lambda A(r) . \tag{11a}$$

Defining P to be the matrix whose columns are comprised of the eigenvectors of Λ with eigenvalues d, we define the quantity

$$B(r) = P^T A(r) . \tag{12}$$

Employing the orthogonality of P ($P^TP = P P^T = 1$), Eq. (11a) becomes

$$dB(r)/dr = D B(r) \tag{13}$$

where $D = P^T \Lambda P$ is diagonal, the diagonal elements being the eigenvalues of Λ. If we rewrite Eq. (13) as

$$dB_m(r)/dr = \sum_n D_{mn} B_n(r) \tag{13a}$$

using $D_{mn} = -|d_m|\delta_{mn}$ (Gerschgorin's Theorem[6]), each element of Eq. (13a) can be solved

$$B_m(r) = \exp(-|d_m|r)\, B_m(0) \quad . \tag{13b}$$

Inverting the transformation Eq. (12), Eq. (13b) yields

$$A_n(r) = \sum_{k,\ell} P_{nk} \exp\left(-|d_k|r\right) P_{k\ell}\, A_\ell(0) . \tag{14}$$

Calculating $A_n(r + R)$ we find

$$A_n(r+R) = \sum_{k,\ell} P_{nk} \exp(-|d_k|R)\, P_{k\ell}\, A_\ell(r) . \tag{15}$$

Note that Eq. (15) is an exact relation, R is not limited in magnitude. Eq. (15) describes how to advance the average correlation amplitude to range r + R knowing its value at range r. In particular, if we know the initial value of A_n, i.e. $A_n(0)$, we can calculate $A_n(r)$ for any range $r > 0$ using Eq. (15).

Numerically, in order to avoid recalculating a double summation, which is time consuming and uses unnecessary storage, we define the matrix $C_{n\ell}(R)$ by

$$C_{n\ell}(R) = \sum_k P_{nk} \exp\left(-|d_k|R\right) P_{k\ell} \quad . \tag{16}$$

This matrix can be calculated once for each roughness setting and stored. Eq. (15) is recast into the form

$$A_n(r+R) = \sum_{\ell} C_{n\ell}(R)\ A_{\ell}(r) \quad . \tag{17}$$

In Eq. (17), C can be regarded as a "translation operator". The quantity of interest here is

$$I(r,z) = \langle|p(r,z,t)|^2\rangle = J(r,z)/r \tag{18}$$

where

$$J(r,z) = \sum_n A_n(r)\ \phi_n^2(z)/\sum_n A_n(r) \quad . \tag{18a}$$

I(r,z) is called the average intensity and does not depend on t within the context of our model. Removing the cylindrical spreading factor, 1/r, J(r,z) shows how the rough surface affects the propagation of the acoustic signal. We make use of the initial condition to find

$$a_n(0) = \left((2\pi/k_0)^{1/2}/4\pi\right)\ \phi_n(z_0) \tag{19a}$$

and thus

$$A_n(0) = \left(1/8\pi k_0\right)\ \phi_n{}^2(z_0) \quad . \tag{19b}$$

Knowing this initial condition completely determines the average intensity.

We now discuss calculation of the normal-mode depth functions, $\phi_n(z)$. The depth functions are calculated by a multiple shooting method. The computer code that we use was developed by L. B.Dozier[7,8] applied to the ocean waveguide. Similar techniques were used before in solving one dimensional radial Schroedinger equations. Boundary conditions are imposed at $z = 0, H$, then the normal mode depth equations are integrated (by a Numerov scheme) from the surface into the deep ocean and simultaneously from the ocean floor upward. The integrations continue to a matching point somewhere in between. In the process of integrating, a trial eigenvalue is used (it is initially estimated using a WKB approach). A comparison of the solution at the matching point is performed to properly match the upward and downward integrations. This also allows one to improve the initial guess made for the eigenvalue for that particular mode. The improvement of the eigenvalue estimate is carried out by an iterative process and continues until convergence to some preset error is reached.

The diagonalization of the matrix Λ and calculation of its eigenvalues are performed by subroutines developed by the Argonne National Laboratory in a program package called EISPACK. The subroutines are called TRED2 and IMTQL2. TRED2 reduces a real symmetric matrix into a symmetric tridiagonal matrix. IMTQL2 takes the tridiagonal matrix and calculates its eigenvalues and eigenvectors. It forms the orthogonal matrix P whose columns are the eigenvectors of Λ.

The above subroutines (TRED2 and IMTQL2) are employed in a computer code to perform the manipulations leading to the construction of Λ and the subsequent implementations of Eqs. (17) and (18). The results of the calculations of the normal modes are read in from an external tape.

IV. APPLICATION TO ARCTIC ICE SCATTERING

A. MODELS

We represent the velocity profile as bilinear:

$$\begin{aligned} c(z)/c_0 &= a + bz \ , \quad 0 \le z \le z_m \\ &= c + dz \ , \quad z_m \le z \le H \ , \end{aligned} \qquad (20)$$

where a = 1, b = 3.48765 x $10^{-5}m^{-1}$, c = 1.006106525 and d = 1.67741 10^{-5} m^{-1}, z_m is the depth at which the two positive gradients b and d meet; z_m = 337.332 m. H is the Arctic Ocean depth, taken to be 4 kilometers.

The multiple shooting method developed to calculate normal modes in the non-ice covered ocean has been adapted to calculate the normal modes for the bilinear profile. Because of the structure of the Arctic velocity profile, there are only RSR modes (RSRBR modes again will not be considered). Therefore the scattering will cause more rapid energy transfer among the RSR modes than in the nonpolar oceans. Thus equipartition is reached sooner as compared with nonpolar ocean scattering.

An important point to be made is that due to the nature of the bilinear profile, it is necessary to use a different initial guess for the eigenvalues from the one used for the canonical profile of the deep sound channel. To estimate the initial eigenvalue we use the fact that the gradients are very small for the bilinear profile and approximate the profile as linear. We then go through the WKB analysis for this initial guess, using the connection formulas, and obtain for our guess,

$$\mu_n^{(0)} = (3b\pi/2k_0)^{2/3}\ (2n + 3/2)^{2/3} \quad . \tag{21}$$

A fundamental difference between Arctic ice scattering and ocean surface scattering is the surface wavenumber spectrum. We take the under-ice surface wavenumber distribution (keel distribution) to be Gaussian[4] with r.m.s. wave height σ and correlation length L:

$$P_1(k) = (\sigma^2 L/\pi^{1/2})\ \exp\ (-k^2 L^2) \quad . \tag{22}$$

The value of L is based on experimental observation made by O. I. Diachok.[9] He measured the order of 10 ice ridges per kilometer. Thus we take L to be of order 100 meters.

The corresponding master equations are

$$dA_n(r)/dr = \sum_{m\neq n} \Gamma_{nm}\left(A_m(r) - A_n(r)\right) \tag{23}$$

where now

$$\Gamma^s_{nm} = \Gamma_{nm} = 2\pi\left(k_0{}^2\sigma^2 L\lambda^4_{nm}/\pi^{1/2}\right)\ \exp\ \left(-\lambda^2_{nm}L^2\right)\ |Z_{nm}|^2 \tag{24}$$

where λ_{nm} ahd Z_{nm} have the same meaning as before.

Before going to the results, we mention the similarities and differences between our formalism and that given by Kryazhev, et al.[1] They assume a fixed boundary, allow for volume attenuation in the ice and bottom, formally consider 2-D transverse phenomena and their integral "transport equation", satisfied by $A_n(r)$, must be calculated at every range point r. In our formalism, the surface can be moving or fixed. The moving surface application allows us to calculate long-range power spectra. We have not yet described dissipative mechanisms in our formalism, but this has in fact been done as explained below. Our formalism is much more efficient when it comes to calculating $A_n(r)$. Finally, they calculate a quantity called the propagation anomaly (PA) related to our transmission loss (TL) by

$$PA = TL + 20 \log (r/r_1) \tag{25}$$

where $r_1 = 1$ yd.

Attenuation due to surface scattering has been studied by Marsh, Schulkin and Kneale[10]. The factor

$$\sin\theta/r_1$$

in the attenuation constant is evaluated for a limiting ray whose vertex depth is 1 km.

Since

$$\sin\theta \sim \theta = \{1-(C_o/C(z))^2\}^{1/2}$$

and

$$C_o/C(z) = (1 + dz)^{-1}$$

with r_1 being determined by ray theory

$(r_1 = 2\theta_o/d,\ \theta_o$ initial ray direction)

it is found that

$$\sin\theta/r_1 = \theta_o/r_1 = d/2\ ,$$

which is independent of r_1. Thus an empirical relation expressing ice scattering per unit distance (dB/km) is obtained, which depends upon the source frequency, f, and the r.m.s. surface roughness, σ :

$$\beta_{MSK} = 9.3373 \times 10^{-6} f^{3/2} \sigma^{8/5}\ \text{dB/km}\ . \qquad (26)$$

For a frequency of 90 Hz with r.m.s. surface roughness of 3 meters, $\beta_{MSK} = .0462$ dB/km. Ref. 1, Fig. 6 (curve 3) indicates that the attenuation is approximately .043 dB/km. Thus to compare the propagation anomaly of Kryazhev et al[1] with the present calculation we need to evaluate:

$$\text{P.A.} = \text{TL} + 20\log_{10}(r/r_1) - 9.3373 \times 10^{-6} f^{3/2}\sigma^{8/5} r\ . \qquad (27)$$

B. RESULTS

Figure 1 describes the quantity $J(r,z)$ (see Eq.(18a) as a function of depth for two particular ranges. The function $J(r,z)$ explicitly indicates the effect of the rough surface (through the amplitude $A_n(r)$). One concludes from this figure that the rough surface serves to redistribute the initial (r = o km) intensity distribution as one steps out to long (r = 200 km) ranges. It is clear from Fig. 1 that the intensity distribution approaches uniformity in the sense that it is no longer possible to distinguish the source position relative to other depths. Figure 2 numerically verifies equipartition (Eq.(8a)) for two particular modes, mode 1 and mode 51. Comparing the approach to equipartition in the deep ocean (Atlantic or Pacific) one notes that the approach to equipartition in the Arctic occurs at a more rapid rate. The reason is that in the Arctic Ocean all modes are RSR modes . Figure 3 displays transmission loss as a function of range for various surface roughnesses. A source with frequency 50 Hz (60 modes are calculated to include all RSR modes) is placed 267 m below the surface. The receiver is at a depth of 260 m. As the roughness of the surface increases, so does transmission loss. For a surface of r.m.s. height 10 meters there is an additional loss at 200 km from the source of 5-6 dB relative to a surface of r.m.s. height 3 meters. Comparison of our theoretical predictions with experimental observations shown by H. W. Kutschale[3] explicitly indicate the need to include surface and bottom loss. If Eq.(26) is employed to account for ice attenuation, then the additional transmission loss corresponding to r.m.s. height of 3 meters at 50 Hz is approximately 20 dB at 1000 km, in closer agreement with experimental observation. The remaining discrepancy can be attributed to the bottom loss not accounted for in our model.

Figure 4 compares the result of our formalism to the work done by Kryazhev,et al[1]. The r.m.s. roughness is 3 meters. The correlation length of the surface flucuations is 50 meters. The source depth is 50 meters and the receiver is located 100 meters below the surface. The source frequency is 90 Hz. The solid line represents the propagation anomoly by our model. The crosses correspond to the work of Kryazhev, et al[1].

ACKNOWLEDGEMENT

This work was supported by the Office of Naval Research under Contract N00014-75-C-0595 with the Lamont-Doherty Geological Observatory of Columbia University and Contract N00014-75-C-0218 with the Courant Institute of Mathematical Sciences of New York University. Professor Daljit S. Ahluwalia made many helpful suggestions during the course of the work.

REFERENCES

1. Kryazhev, F. I., V. M. Kudryashov and N. A. Petrov, "Propagation of Low-Frequency Sound Waves in a Waveguide with Irregular Boundaries," Sov. Phys. Acoust. 22, No. 3, 211 (1976).

2. Tappert, F. D., Lecture Notes in Physics, No. 70, Wave Propagation and Underwater Acoustics, ed. J. B. Keller and J. S. Papadakis, Springer-Verlag, Berlin, 1977.

3. Kutschale, H. W., "Rapid Computation by Wave Theory of Propagation Loss in the Arctic Ocean," Tech. Rep. No. 8, Lamont-Doherty Geological Observatory, Columbia University, March, 1973.

4. Hibler, W. D. III, W. F. Weeks, S. J. Mock, "Statistical Aspects of Sea-Ice Ridge Distributions," J. Geo. Res. 77, 5954 (1972).

5. Beilis, A., "Coupled Mode Analysis of Multiple Rough Surface Scattering," Ph.D. Dissertation, N.Y.U., March, 1978.

6. Kreysig, E., Advanced Engineering Mathematics, 2nd Ed., J. Wiley, New York, 1962, Chap. 7, p. 422.

7. Dozier, L. B., "Calculation of Normal Modes in a Stratified Ocean," unpublished report.

8. Dozier, L. B. and F. D. Tappert, "Statistics of Normal Mode Amplitudes in a Random Ocean.I.Theory," J. Acoust. Soc. Amer. 63, 353(1978).

9. Diachok, O. I., "Effects of Sea-Ice Ridges on Sound Propagation in the Arctic Ocean," J. Acoust. Soc. Amer., 59, 1110(1976).

10. Marsh, H. W., M. Schulkin, and S. G. Kneale, "Scattering of Underwater Sound by the Sea Surface," J. Acoust. Soc. Am., 33, 334 (1961).

FIGURE CAPTIONS

Figure 1. Mean intensity at two specific ranges (0, 200 km) as a function of depth, with the cylindrical spreading factor removed. The source frequency is 50 Hz, the r.m.s. roughness is 10 m., the source depth is 267 m.

Figure 2. Verifies the approach to equipartition of energy (Eq.8a). Same frequency,source depth and r.m.s. roughness as for Figure 1.

Figure 3. Transmission loss as a function of range for 3 r.m.s. roughnesses. Source depth 267 m., receiver depth 260 m, frequency 50 Hz.

Figure 4. Propagation anomaly (see Eqs. 25 and 27). Source frequency is 90 Hz, source depth is 50 m. r.m.s. roughness 3 m, correlation length 50 m., receiver depth 100 m.

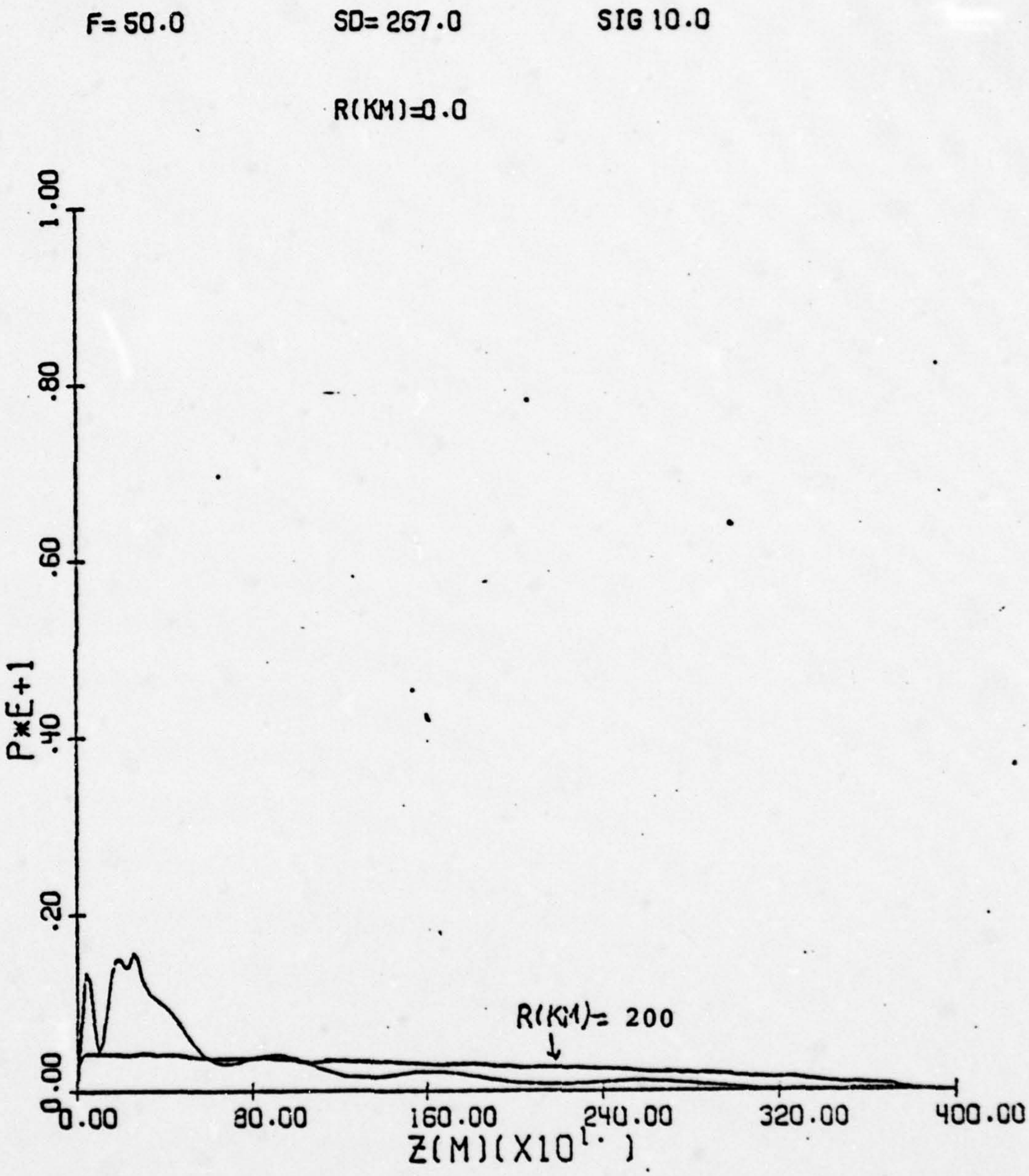

F= 50.0
SD= 267.0
SIG 10.0
R(KM)=0.0
1.00
.80
.60
.40
.20
0.00
P*E+1
R(KM)= 200
0.00
80.00
160.00
240.00
320.00
400.00
Z(M)(X10[··)

FIGURE 1

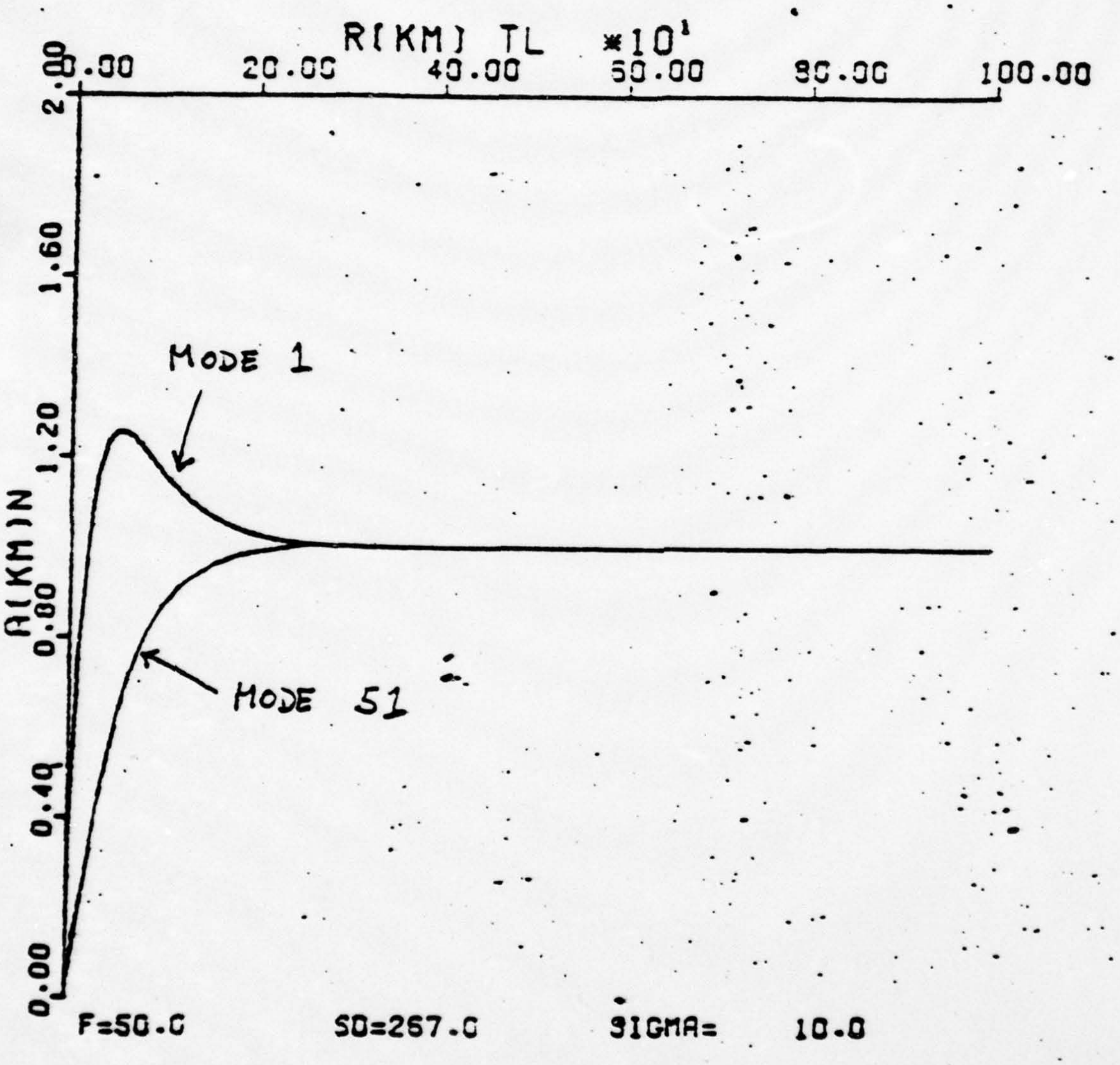

R(KM) TL *10¹
0.00
20.00
40.00
60.00
80.00
100.00
A(KM)N
2.00
1.60
1.20
0.80
0.40
0.00
MODE 1
MODE 51
F=50.0
SD=267.0
SIGMA= 10.0

FIGURE 2

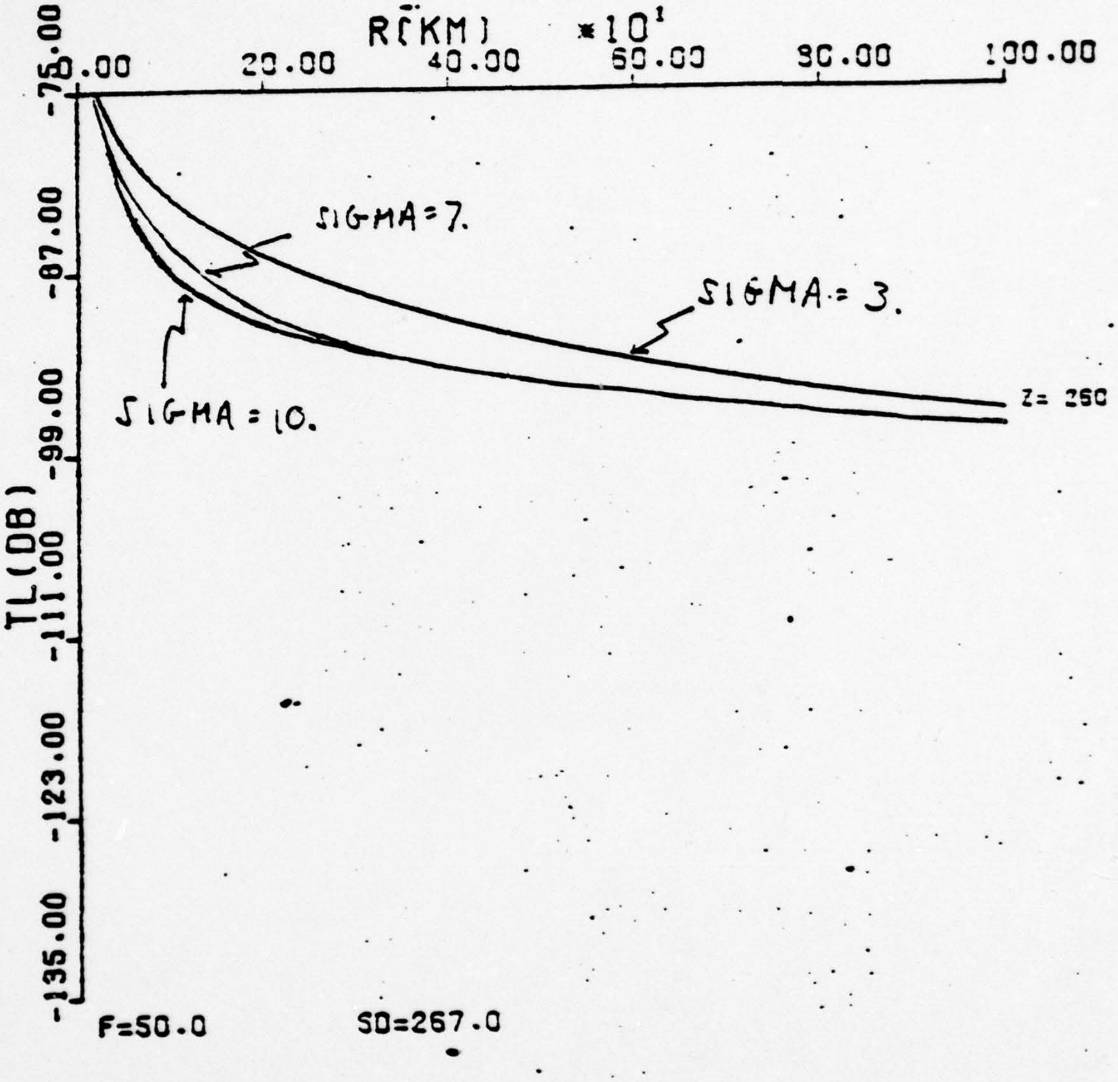

R(KM) *10^1
0.00
20.00
40.00
60.00
80.00
100.00
-75.00
-87.00
-99.00
-111.00
-123.00
-135.00
TL(DB)
SIGMA=7.
SIGMA=3.
SIGMA=10.
Z= 250
F=50.0
SD=267.0

FIGURE 3

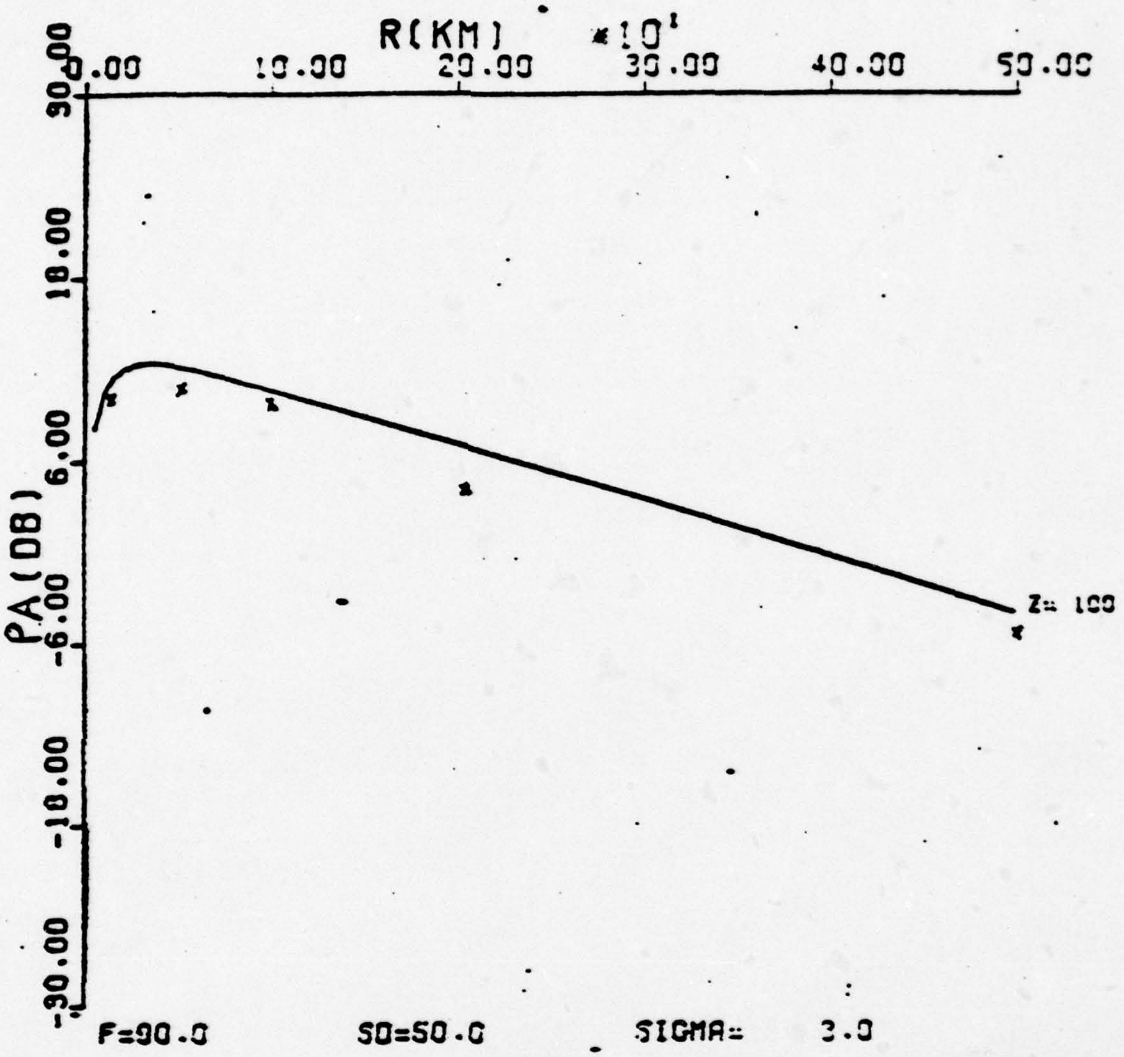

R(KM) *10^1
0.00
10.00
20.00
30.00
40.00
50.00
PA(DB)
30.00
18.00
6.00
-6.00
-18.00
-30.00
Z= 100
F=90.0
SO=50.0
SIGMA= 3.0

FIGURE 4

APPENDIX A

DOCUMENTATION OF WIND AND PROGRAM LISTING

$$dA_n(r)/dr = \sum_{m \neq n} \Gamma^s_{nm}\left(A_m(r) - A_n(r)\right)$$

GAMA

Matrix in master equations (meter)$^{-1}$

P

Matrix whose columns are the eigenvectors of GAMA

DG

Elements are the eigenvalues of GAMA

XKPA

Elements are the eigenvalues corresponding to the normal-mode depth functions.

PHIN, PHIM

Normal-mode depth functions

ZZ

Depth variable, meters

AX

Amplitudes, $A_n(r)$

POWR

Used to calculate various quantities, such as, intensity, transmission loss, equipartition, etc.

XR

Range variable in kilometers

CO

Sound speed at ocean surface, meters/sec.

NM

Number of modes

DELTR

Range increment in meters

NR

Number of points in total range (NR-1 is the number of divisions of total range)

SD

Source depth in meters

KO

Acoustic wavenumber, meters^{-1} (Real)

SIGMA

r.m.s. surface roughness, meters

CL

Surface roughness correlation length, meters

FREQ

Acoustic source frequency, Hz

ALPHA

Ice attenuation constant (dB/km)

```
/
      PROGRAM WIND(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9,TAPE10)
C     THIS PROGRAM PLOTS LOG OF RATIO OF INTENSITY TO POINT
C     SOURCE INTENSITY AT 1 METER
C     PLOTS RANGE TIMES INTENSITY
C      PHI IS TO BE IN RADIANS
      COMMON/A/FREQ,SD,SIGMA
      REAL KOSQ,KO
      COMMON GAMA(114,114),P(114,114),DG(114),XKPA(114),DN(114),DE(114),
     1PHIN(201),PHIM(201),ZZ(201),E(114),AX(101,114),POWR(101),XR(101)
      DIMENSION PHINB(114,201)
      DATA HTL/8HR(KM)   /,VTL/10H  TL(DB)  /,VTL1/7HA(KM)*N/
     +,VTL2/7H P*E+1 /,HTL3/4HZ(M)/,VTL3/6HP/PMAX/,
     +HTL4/3H N /,VTL4/8H-LOG/DG//
      NAMELIST/VAL/SIGMA,THRESH,CL
      DATA CO/1434./,PI/3.1415926535/,NM/114/,NPREV,NMM/-2,0/
      DATA DELTR/1.5E+4/,NR/101/,SD/50./,BETA/1.6774E-5/
      REWIND 9
 5    NPREV=NPREV+NMM+2
C SKIP OVER LAST (WAVENO) RECORD OF PREV FREQUENCY
      IF(NPREV.GT.0) READ(9)
      READ(9)B,H,NZ,KOSQ,NMM
      IF(EOF(9))77,7
7     FREQ=CO/          SQRT(KOSQ)
      IF(NM.GT.NMM)GO TO 76
C SKIP OVER THE EIGENFUNCTIONS
      NZ1=NZ-1
      KO=SQRT(KOSQ)
      DO 40 I=1,NMM
40    READ(9)
      READ(9)(XKPA(I),I=1,NM)
      DO 85 I=1,NM
  85  XKPA(I)=XKPA(I)**2/(2.*KO)
      DZ=H/NZ1
      DO 50 I=1,NZ
50    ZZ(I)=(I-1)*DZ
4     READ VAL
      IF(EOF(5))77,51
51    PRINT VAL
      IF(PHI.EQ.90.)PHI=ASIN(1.)
      ALPHA=-9.3373E-6*FREQ**(1.5)*SIGMA**(1.6)
      NMM1=NM-1
      B4=1./(1.+BETA*H)**3
      DO 2 N=1,NMM1
      REWIND 9
      IF(NPREV.LE.0) GO TO 9
      DO 8 I=1,NPREV
8     READ(9)
9     DO 10 I=1,N
10    READ(9)
      READ(9) PHIN
      N1=N+1
      DO 3 M=N1,NM
      X1=XKPA(N)-XKPA(M)
      X=ABS(X1)
      XX=X**2
      READ(9)PHIM
      SUM=0.5*PHIN(NZ)*PHIM(NZ)*H
      SUMA=0.5*PHIN(NZ)*PHIM(NZ)*B4
      DO 60 I=2,NZ1
      A4=1./(1.+BETA*ZZ(I))**3
      SUMA=SUMA+PHIN(I)*PHIM(I)*A4
60    SUM=SUM+PHIN(I)*PHIM(I)*ZZ(I)
      ZMN=SUM*DZ
      ZMN1=SUMA*DZ
C     WRITE(6,1000)N,M,ZMN
C1000 FORMAT(1X,*ZMN(*,I3,*,*,I3,*)=*,E11.4)
      Z2=ZMN**2
      R=XX*CL**2
      IF(R.LE.1.E-8)GO TO 22
      IF(R.GE.350.)GO TO 23
      DD=EXP(-R)
      GO TO 12
  22  DD=1.-R
      GO TO 12
  23  DD=0.
  12  GAMA(N,M)=CL*SIGMA**2*DD*Z2*XX**2*KOSQ/SQRT(PI)
      GAMA(N,M)=(2.*PI)*GAMA(N,M)*(1.-(BETA*ZMN1/(XX*ZMN)))**2
      GAMA(M,N)=GAMA(N,M)
3     CONTINUE
2     CONTINUE
      DO 57 I=1,NM
      GSUM=0.
      DO 58 J=1,NM
      IF(J.EQ.I)GO TO 58
      GSUM=GSUM+GAMA(J,I)
 58   CONTINUE
 57   GAMA(I,I)=-GSUM
C     DO 80 I=1,NM
C  80 WRITE(6,1060)(GAMA(I,J),J=1,I)
C **********CALCULATES COHERENT DECAY LENGTHS***********
      CALL PLOTSBL(275,9HA. BEILIS)
      DO 15 I=1,NM
   15 DE(I)=-ALOG10(ABS(GAMA(I,I)))
      DO 16 J=1,NM
   16 DN(J)=FLOAT(J)
      EMAX=DE(1)
      DO 115 I=1,NM
  115 IF(DE(I).GE.EMAX)EMAX=DE(I)
      EMIN=DE(1)
      DO 116 J=1,NM
  116 IF(DE(J).LE.EMIN)EMIN=DE(J)
      NH4=3
      NV4=8
      XI=PI
      CALL GFX(DN,DE,NM,DN(1),DN(NM),4.,6.,XI,
     +DE(NM),HTL4,NH4,VTL4,NV4)
      DO 17 I=1,NM
   17 DE(I)=0.
      CALL FRAME
C ************************************* ******
      GMAX=0.
      NMM1=NM-1
      DO 55 I=1,NMM1
      JMIN=I+1
      DO 56 J=JMIN,NM
      TEMP=ABS(GAMA(J,I))
      IF(TEMP.GT.GMAX)GMAX=TEMP
56    CONTINUE
55    CONTINUE
      WRITE(6,600)GMAX
600   FORMAT(*0 GMAX=*,E11.4)
C MUST PRINT GAMA ARRAY NOW SINCE MAY BE DESTROYED BY TRED2
C     WRITE(6,1062)
C 1062 FORMAT(*0THE LOWER TRIANGLE OF THE GAMMA MATRIX FOLLOWS*)
      DO 571 I=1,NM
      DO 581 J=1,NM
      IF(ABS(GAMA(J,I)).LT.THRESH*GMAX)GAMA(J,I)=0.
581   CONTINUE
571   CONTINUE
```

```
      CALL TRED2(NM,NM,GAMA,DG,E,P)
      CALL IMTQL2(NM,NM,DG,E,P,IERR)
      IF(IERR.NE.0)GO TO 500
C     WRITE(6,1059)FREQ
C1059  FORMAT(*OTHESE ARE THE EIGENVECTORS CORRESPONDING TO ACOUSTIC
C    1FREQUENCY *,F6.2)
C     DO 70 I=1,NM
C70    WRITE(6,1060)(P(J,I),J=1,NM)
1060  FORMAT(*0*,10E11.4)
      WRITE(6,2223)FREQ,NM,SD
2223  FORMAT(* FREQ=*,F6.2,* # MODES=*,I4,* SDEPTH=*,F7.3)
      WRITE(6,1061)
1061  FORMAT(*0THE EIGENVALUES ARE *)
      WRITE(6,1060)(DG(I),I=1,NM)
      DO 18 I=1,NM
      DE(I)=-ALOG10(ABS(DG(I)))
  18  IF(I.EQ.NM)DE(I)=DE(I-1)
      NH4=3
      NV4=8
      XI=PI
      CALL GFX(DN,DE,NM,DN(1),DN(NM),4.,6.,XI,
     +DE(NM),HTL4,NH4,VTL4,NV4)
      CALL FRAME
      DO 101 I=1,NM
      DO 101 K=1,NM
      SUMB=0.
      DO 103 J=1,NM
      X1=DG(J)*DELTR
      IF(ABS(X1).GE.675.84)GO TO 103
      Y1=P(I,J)*P(K,J)*EXP(X1)
      SUMB=SUMB+Y1
103   CONTINUE
C     BX IS REPLACED BY GAMA TO MIN. SPACE USED.
101   GAMA(I,K)=SUMB
C     WRITE(6,1060)(GAMA(I,K),K=1,NM)
      NZX=INT(SD/DZ)+1
      W=SD/DZ-FLOAT(NZX)+1.
      REWIND 9
      READ(9)
      SUMC=0.
      DO 20 J=1,NM
      READ(9)PHIN
      PHINT=W*PHIN(NZX+1)+(1.-W)*PHIN(NZX)
      AX(1,J)=PHINT**2
20    SUMC=SUMC+PHINT**2
      CONST=1./SUMC
      WRITE(6,3333)CONST
 3333 FORMAT(* CONST=*,E11.4)
C     WRITE(6,3334)AX(1,1),AX(1,50),AX(1,100)
      DO 30 I=1,NM
30    AX(1,I)=AX(1,I)*CONST
C     WRITE(6,3334)AX(1,1),AX(1,50),AX(1,100)
C3334 FORMAT(* AX(1,1,50,100)=*,3E11.4)
      DO 82 L=2,NR
      LM1=L-1
      DO 81 I=1,NM
      SUMA=0.
      DO 83 K=1,NM
83    SUMA=SUMA+GAMA(I,K)*AX(LM1,K)
      AX(L,I)=SUMA
C     IF(MOD(L,10).EQ.1)WRITE(6,2222)LM1,I,AX(L,I)
C222  FORMAT(* AX(*,I4,*,*,I4,*)=*,E11.4)
81    CONTINUE
82    CONTINUE
C     DO 701 L=2,NR
C     ENCON=0.
```

```
C      DO 711 I=1,NM
C711   ENCON=ENCON+AX(L,I)
C701    WRITE(6,9999)ENCON,L
C9999  FORMAT(* ENCON=*,E11.4,* RANGE=40TIMES*,I4,*KM*)
       DO 605 J=1,NR
 605   XR(J)=FLOAT(J-1)*DELTR/1.E+3
       NH=8
       REWIND 9
       READ(9)
       DO 27 N=1,NM
 27    READ(9)(PHINB(N,I),I=1,NZ)
       XA=2.*ALOG10(12./13.)
       XB1=ALOG10(2.*PI/(K0*CONST))
       DO 501 M=1,6
       DOD=100.*FLOAT(M-1)+100.
       NZY=INT(DOD/DZ)+1
       W1=DOD/DZ-FLOAT(NZY)+1.
       SUMP1=0.
       DO 817 N=1,NM
       PHINA=W1*PHINB(N,NZY+1)+(1.-W1)*PHINB(N,NZY)
 817   SUMP1=SUMP1+AX(1,N)*PHINA**2
       POWR(1)=SUMP1
C      WRITE(6,503)POWR(1)
C503   FORMAT(* POWER(1,Z)=*,E11.4)
       POWR(1)=10.*(ALOG10(POWR(1)*13./12.)+XA+XB1)
       WRITE(6,98)POWR(1)
   98 FORMAT(1X,*LOG(I/S)(1)=*,E11.4)
       IF(ABS(POWR(1)).LT.80.)POWR(1)=-80.
       IF(ABS(POWR(1)).GT.130.)POWR(1)=-130.
       DO 63 I=2,NR
       SUMP=0.
       DO 807 N=1,NM
       PHINX=W1*PHINB(N,NZY+1)+(1.-W1)*PHINB(N,NZY)
807    SUMP=SUMP+AX(I,N)*PHINX**2
       POWR(I)=SUMP
       POWR(I)=10.*(ALOG10(SUMP/(XR(I)*1.E+3))+XA+XB1)
C      POWR(I)=POWR(I)+ALPHA*XR(I)
C      POWR(I)=POWR(I)+20.*ALOG10(XR(I)*1.E+3*13./12.)
       WRITE(6,99)I,POWR(I)
   99 FORMAT(1X,*LOG(I/S)(*,I4,*)=*,E11.4)
       IF(ABS(POWR(I)).GT.80.)POWR(I)=-80.
       IF(ABS(POWR(I)).GT.130.)POWR(I)=-130.
C      IF(ABS(POWR(I)).GT.15.)POWR(I)=15.
C      WRITE(6,2323)I,DOD,SUMP,POWR(I)
C2323 FORMAT(* P(*,I2,*,*,F7.2,*)=*,E11.4,*  LOG(P)=*,E11.4)
 63    CONTINUE
C      YMAX=0.
C      DO 827 I=1,NR
C827   IF(POWR(I).GT.YMAX)YMAX=POWR(I)
C      YMIN=YMAX
C      DO 837 I=1,NR
C837   IF(POWR(I).LT.YMIN)YMIN=POWR(I)
       IF(M.GT.1)NH=-8
       XQ=0.3
       XMQ=0.
       CALL YFX(XR,POWR,NR,XR(1),XR(NR),-130.,-80.,HTL,NH,VTL,10,M,XQ,
      +XMQ)
 501   CONTINUE
       CALL FRAME
       NH=8
       DO 601 M=1,6
       DOD=100.*FLOAT(M-1)+100.
       NZY=INT(DOD/DZ)+1
       W1=DOD/DZ+1.-FLOAT(NZY)
       SUMP1=0.
       DO 917 N=1,NM
      PHINA=W1*PHINB(N,NZY+1)+(1.-W1)*PHINB(N,NZY)
 917  SUMP1=SUMP1+AX(1,N)*PHINA**2
      POWR(1)=SUMP1*1.E+2
C     WRITE(6,603)DOD,POWR(1)
C603  FORMAT(* P(1,*,F7.2,*)=*,E11.4)
      DO 73 I=2,NR
      SUMP=0.
      DO 907 N=1,NM
      PHINX=W1*PHINB(N,NZY+1)+(1.-W1)*PHINB(N,NZY)
 907  SUMP=SUMP+AX(I,N)*PHINX**2
      POWR(I)=SUMP*1.E+2
C     WRITE(6,4343)I,DOD,SUMP
C4343 FORMAT(* P(*,I3,*,*,F7.2,*)=*,E11.4)
  73  CONTINUE
C     YMX=0.
C     DO 847 I=1,NR
C847  IF(POWR(I).GT.YMX)YMX=POWR(I)
C     WRITE(6,4000)YMX
C4000 FORMAT(* PMAX=*,E11.4)
C     DO 857 I=1,NR
C857  POWR(I)=POWR(I)/YMX
      IF(M.GT.1)NH=-8
      XQ=6.5
      XMQ=0.
      CALL YFX(XR,POWR,NR,XR(1),XR(NR),0.,1.,HTL,NH,VTL2,7,M,XQ,XMQ)
 601  CONTINUE
      CALL FRAME
      XMQ=1.
      DOD=500.
      NZY=INT(DOD/DZ)+1.
      WX=DOD/DZ+1.-FLOAT(NZY)
      DO 791 N=1,NM,15
      NH=8
C     PHINX=WX*PHINB(N,NZY+1)+(1.-WX)*PHINB(N,NZY)
      DO 792 I=1,NR
      POWR(I)=AX(I,N)*FLOAT(NM)
      IF(POWR(I).GT.2.)WRITE(6,26)I,N,POWR(I)
   26 FORMAT(1X,*POWR(*,I3,*,*,I3,*)=*,E11.4)
 792  IF(POWR(I).GT.2.)POWR(I)=2.
      CALL YFX(XR,POWR,NR,XR(1),XR(NR),0.,2.,HTL,NH,VTL1,7,N,XQ,XMQ)
  791 CALL FRAME
      DO 793 I=1,NR,10
      XI=FLOAT(I-1)*DELTR/1.E+3
      IX=XI
      IF(I.EQ.NR)GO TO 666
      IF(IX.GT.450.AND.(MOD(IX,300).NE.0))GO TO 793
  666 DO 794 J=1,NZ
      SUMX=0.
      DO 795 N=1,NM
  795 SUMX=SUMX+AX(I,N)*PHINB(N,J)**2
  794 PHIN(J)=SUMX
      YMIX=0.
      DO 89 J=1,NZ
   89 IF(PHIN(J).GE.YMIX)YMIX=PHIN(J)
      DO 91 J=1,NZ
   91 PHIN(J)=PHIN(J)*1.E+2
      NH3=4
      NV3=6
      CALL GFX(ZZ,PHIN,NZ,ZZ(1),ZZ(NZ),0.,1.,XI,YMIX,
     +HTL3,NH3,VTL2,NV3)
      CALL FRAME
  793 CONTINUE
      CALL PLOT(0.0,0.0,999)
      GO TO 4
500   WRITE(6,1050)IERR
1050  FORMAT(*0IERR= *,I20)
```

```
      IERR=IERR-1
      IF(IERR.GE.1.AND.IERR.LT.251)WRITE(6,1060)(DG(I),I=1,IERR)
      GO TO 5
76     WRITE(6,1080)NMM
1080  FORMAT(*0ERROR,ONLY*,I5,*MODES ON DISK*)
77    STOP
      END
C
      SUBROUTINE IMTQL2(NM,N,D,E,Z,IERR)
      INTEGER I,J,K,L,M,N,II,NM,MML,IERR
      REAL D(N),E(N),Z(NM,N)
      REAL B,C,F,G,P,R,S,MACHEP
      MACHEP = 2.**(-47)
C
      IERR = 0
      IF (N .EQ. 1) GO TO 1001
C
      DO 100 I = 2, N
  100 E(I-1) = E(I)
C
      E(N) = 0.0
C
      DO 240 L = 1, N
         J = 0
C     ********** LOOK FOR SMALL SUB-DIAGONAL ELEMENT **********
  105    DO 110 M = L, N
            IF (M .EQ. N) GO TO 120
            IF (ABS(E(M)) .LE. MACHEP * (ABS(D(M)) + ABS(D(M+1))))
     X         GO TO 120
  110    CONTINUE
C
  120    P = D(L)
         IF (M .EQ. L) GO TO 240
         IF (J .EQ. 30) GO TO 1000
         J = J + 1
C     ********** FORM SHIFT **********
         G = (D(L+1) - P) / (2.0 * E(L))
         R = SQRT(G*G+1.0)
         G = D(M) - P + E(L) / (G + SIGN(R,G))
         S = 1.0
         C = 1.0
         P = 0.0
         MML = M - L
C     ********** FOR I=M-1 STEP -1 UNTIL L DO -- **********
         DO 200 II = 1, MML
            I = M - II
            F = S * E(I)
            B = C * E(I)
            IF (ABS(F) .LT. ABS(G)) GO TO 150
            C = G / F
            R = SQRT(C*C+1.0)
            E(I+1) = F * R
            S = 1.0 / R
            C = C * S
            GO TO 160
  150       S = F / G
            R = SQRT(S*S+1.0)
            E(I+1) = G * R
            C = 1.0 / R
            S = S * C
  160       G = D(I+1) - P
            R = (D(I) - G) * S + 2.0 * C * B
            P = S * R
            D(I+1) = G + P
            G = C * R - B
C     ********** FORM VECTOR **********
```

```
            DO 180 K = 1, N
               F = Z(K,I+1)
               Z(K,I+1) = S * Z(K,I) + C * F
               Z(K,I) = C * Z(K,I) - S * F
  180       CONTINUE
C
  200    CONTINUE
C
         D(L) = D(L) - P
         E(L) = G
         E(M) = 0.0
         GO TO 105
  240 CONTINUE
C     *********** ORDER EIGENVALUES AND EIGENVECTORS ***********
      DO 300 II = 2, N
         I = II - 1
         K = I
         P = D(I)
C
         DO 260 J = II, N
            IF (D(J) .GE. P) GO TO 260
            K = J
            P = D(J)
  260    CONTINUE
C
         IF (K .EQ. I) GO TO 300
         D(K) = D(I)
         D(I) = P
C
         DO 280 J = 1, N
            P = Z(J,I)
            Z(J,I) = Z(J,K)
            Z(J,K) = P
  280    CONTINUE
C
  300 CONTINUE
C
      GO TO 1001
C     *********** SET ERROR -- NO CONVERGENCE TO AN
C                 EIGENVALUE AFTER 30 ITERATIONS ***********
 1000 IERR = L
      WRITE(6,2050)L
 2050 FORMAT(*OL= *,I20)
 1001 RETURN
C     *********** LAST CARD OF IMTQL2 ***********
      END
C
      SUBROUTINE TRED2(NM,N,A,D,E,Z)
C
      INTEGER I,J,K,L,N,II,NM,JP1
      REAL A(NM,N),D(N),E(N),Z(NM,N)
      REAL F,G,H,HH,SCALE
      REAL SQRT,ABS,SIGN
C
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRED2,
C     NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 212-226(1971).
C
C     THIS SUBROUTINE REDUCES A REAL SYMMETRIC MATRIX TO A
C     SYMMETRIC TRIDIAGONAL MATRIX USING AND ACCUMULATING
C     ORTHOGONAL SIMILARITY TRANSFORMATIONS.
C
C     ON INPUT-
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
```

```
C          DIMENSION STATEMENT.
C
C        N IS THE ORDER OF THE MATRIX.
C
C        A CONTAINS THE REAL SYMMETRIC INPUT MATRIX.  ONLY THE
C          LOWER TRIANGLE OF THE MATRIX NEED BE SUPPLIED.
C
C     ON OUTPUT-
C
C        D CONTAINS THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX.
C
C        E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE TRIDIAGONAL
C          MATRIX IN ITS LAST N-1 POSITIONS.  E(1) IS SET TO ZERO.
C
C        Z CONTAINS THE ORTHOGONAL TRANSFORMATION MATRIX
C          PRODUCED IN THE REDUCTION.
C
C        A AND Z MAY COINCIDE.  IF DISTINCT, A IS UNALTERED.
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW.
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C     ------------------------------------------------------------------
C
      DO 100 I = 1, N
C
         DO 100 J = 1, I
           Z(I,J) = A(I,J)
  100 CONTINUE
C
      IF (N .EQ. 1) GO TO 320
C     ********** FOR I=N STEP -1 UNTIL 2 DO -- **********
      DO 300 II = 2, N
         I = N + 2 - II
         L = I - 1
         H = 0.0
         SCALE = 0.0
         IF (L .LT. 2) GO TO 130
C     ********** SCALE ROW (ALGOL TOL THEN NOT NEEDED) **********
         DO 120 K = 1, L
  120    SCALE = SCALE + ABS(Z(I,K))
C
         IF (SCALE .NE. 0.0) GO TO 140
  130    E(I) = Z(I,L)
         GO TO 290
C
  140    DO 150 K = 1, L
           Z(I,K) = Z(I,K) / SCALE
           H = H + Z(I,K) * Z(I,K)
  150    CONTINUE
C
         F = Z(I,L)
         G = -SIGN(SQRT(H),F)
         E(I) = SCALE * G
         H = H - F * G
         Z(I,L) = F - G
         F = 0.0
C
         DO 240 J = 1, L
           Z(J,I) = Z(I,J) / (SCALE * H)
           G = 0.0
C     ********** FORM ELEMENT OF A*U **********
           DO 180 K = 1, J
  180      G = G + Z(J,K) * Z(I,K)
C
           JP1 = J + 1
```

```
              IF (L .LT. JP1) GO TO 220
C
              DO 200 K = JP1, L
  200         G = G + Z(K,J) * Z(I,K)
C     *********** FORM ELEMENT OF P **********
  220         E(J) = G / H
              F = F + E(J) * Z(I,J)
  240     CONTINUE
C
          HH = F / (H + H)
C     *********** FORM REDUCED A **********
          DO 260 J = 1, L
              F = Z(I,J)
              G = E(J) - HH * F
              E(J) = G
C
              DO 260 K = 1, J
                  Z(J,K) = Z(J,K) - F * E(K) - G * Z(I,K)
  260     CONTINUE
C
          DO 280 K = 1, L
  280     Z(I,K) = SCALE * Z(I,K)
C
  290     D(I) = H
  300 CONTINUE
C
  320 D(1) = 0.0
      E(1) = 0.0
C     *********** ACCUMULATION OF TRANSFORMATION MATRICES **********
      DO 500 I = 1, N
          L = I - 1
          IF (D(I) .EQ. 0.0) GO TO 380
C
          DO 360 J = 1, L
              G = 0.0
C
              DO 340 K = 1, L
  340         G = G + Z(I,K) * Z(K,J)
C
              DO 360 K = 1, L
                  Z(K,J) = Z(K,J) - G * Z(K,I)
  360     CONTINUE
C
  380     D(I) = Z(I,I)
          Z(I,I) = 1.0
          IF (L .LT. 1) GO TO 500
C
          DO 400 J = 1, L
              Z(I,J) = 0.0
              Z(J,I) = 0.0
  400     CONTINUE
C
  500 CONTINUE
C
      RETURN
C     *********** LAST CARD OF TRED2 **********
      END
      SUBROUTINE YFX(X,Y,N,XMIN,XMAX,YMIN,YMAX,HTL,NH,VTL,NV,M,XQ,XMQ)
      DIMENSION TEXT(7),X(N),Y(N)
      COMMON/A/FREQ,SD,SIGMA
      DATA (TEXT(I),I=1,7)/10HOD= 100.00,10HOD= 200.00,
     +10HOD= 300.00,10HOD= 400.00,10HOD= 500.00,10HOD= 600.00,
     +10HOD= 700.00/
      WRITE(6,2014)(X(I),I=1,N)
      WRITE(6,2014)(Y(J),J=1,N)
 2014 FORMAT(1X,10E11.4)
```

```
      IF(NH.LE.0)GO TO 10
      DX=(XMAX-XMIN)/5.
      DY=(YMAX-YMIN)/5.
      XB=0.5
      IF(XQ.EQ.0.3)XB=5.5
      IF(XB.EQ.0.5)NH=-NH
      CALL AXIS(0.5,XB,HTL,NH,5.,0.,XMIN,DX)
      CALL AXIS(0.5,0.5,VTL,NV,5.,90.,YMIN,DY)
      CALL SYMBOL(0.5,XQ,.1,3H F=,0.0,3)
      CALL NUMBER(0.8,XQ,.1,FREQ,0.0,1)
      CALL SYMBOL(2.0,XQ,.1,3HSD=,0.0,3)
      CALL NUMBER(2.3,XQ,.1,SD,0.0,1)
      CALL SYMBOL(3.5,XQ,.1,6HSIGMA=,0.0,3)
      CALL NUMBER(3.8,XQ,.1,SIGMA,0.0,1)
      IF(XMQ.EQ.1.)Z=M
      IF(XMQ.EQ.1.)CALL SYMBOL(3.5,6.0,.1,5HMODE ,0.0,5)
      IF(XMQ.EQ.1.)CALL NUMBER(4.0,6.0,.1,Z,0.0,1)
 10   CALL LINE(X,Y,N,1,0,0,XMIN-DX/2.,DX,YMIN-DY/2.,DY)
      ZETA=(Y(N)-YMIN)/DY+.5
      IF(Y(N).LT.0.)ZETA=XB-ABS(Y(N)-YMAX)/DY
C     ZETA DEPENDS ON CHOICE OF AXIS
C     RE-ADJUST AS REQUIRED
      IF(XMQ.EQ.1.)GO TO 12
      CALL SYMBOL(X(N)/DX+.6,ZETA,.08,TEXT(M),0.,10)
 12   RETURN
      END
      SUBROUTINE GFX(X,Y,N,XMIN,XMAX,YMIN,YMAX,XI,YMIX,
     +HTL3,NH3,VTL3,NV3)
      DIMENSION X(N),Y(N)
      COMMON/A/FREQ,SD,SIGMA
      WRITE(6,4014)(X(I),I=1,N)
      WRITE(6,4014)(Y(J),J=1,N)
4014  FORMAT(1X,10E11.4)
      IF(XI.NE.PI)WRITE(6,4024)XI,YMIX
4024  FORMAT(1X,*PMAX(*,F6.1,* KM)=*,E11.4)
      IF(NH3.LE.0)GO TO 10
      DX=(XMAX-XMIN)/5.
      DY=(YMAX-YMIN)/5.
      CALL AXIS(0.5,0.5,HTL3,-NH3,5.,0.,XMIN,DX)
      CALL AXIS(0.5,0.5,VTL3,NV3,5.,90.,YMIN,DY)
      CALL SYMBOL(0.5,6.5,.1,3H F=,0.0,3)
      CALL NUMBER(0.8,6.5,.1,FREQ,0.0,1)
      CALL SYMBOL(2.0,6.5,.1,3HSD=,0.0,3)
      CALL NUMBER(2.3,6.5,.1,SD,0.0,1)
      CALL SYMBOL(3.5,6.5,.1,6HSIGMA=,0.0,3)
      CALL NUMBER(3.8,6.5,.1,SIGMA,0.0,1)
      IF(XI.NE.PI)CALL SYMBOL(2.0,6.0,.1.6HR(KM)=,0.0,6)
      IF(XI.NE.PI)CALL NUMBER(2.5,6.0,.1.XI,0.0,1)
 10   CALL LINE(X,Y,N,1,0,0,XMIN-DX/2.,DX,YMIN-DY/2.,DY)
      RETURN
      END
 -END OF FILE-
```

APPENDIX B

DOCUMENTATION OF NORMAL MODE PROGRAM FOR THE BILINEAR PROFILE AND PROGRAM LISTING

P
Normal modes ϕ_n as a function of depth

U
Eigenvalues μ_n

STK
Related to eigenvalues via STK = $k_o \sqrt{1-\mu_n}$
where k_o is the source wavenumber

ZM Depth coordinate (variable)

ZMAT
Depth coordinate at which up and down integrations meet

ZMEET
Depth coordinate where the two velocity gradients meet

EN
Index of refraction, $n(z)$, as a function of depth

VM
The potential, $1-n^2(z)$, in the normal-mode equation

DELZ
Depth increment

KO
Source wavenumber

KOSQ
Source wavenumber squared

BC
Boundary condition value for eigenfunction

SLOPE
Slope of eigenfunction at the boundary

TPT
Turning point where eigenfunctions turn from oscillatory (exponentially decay) to exponentially decaying (oscillatory)

H
Ocean depth (m)

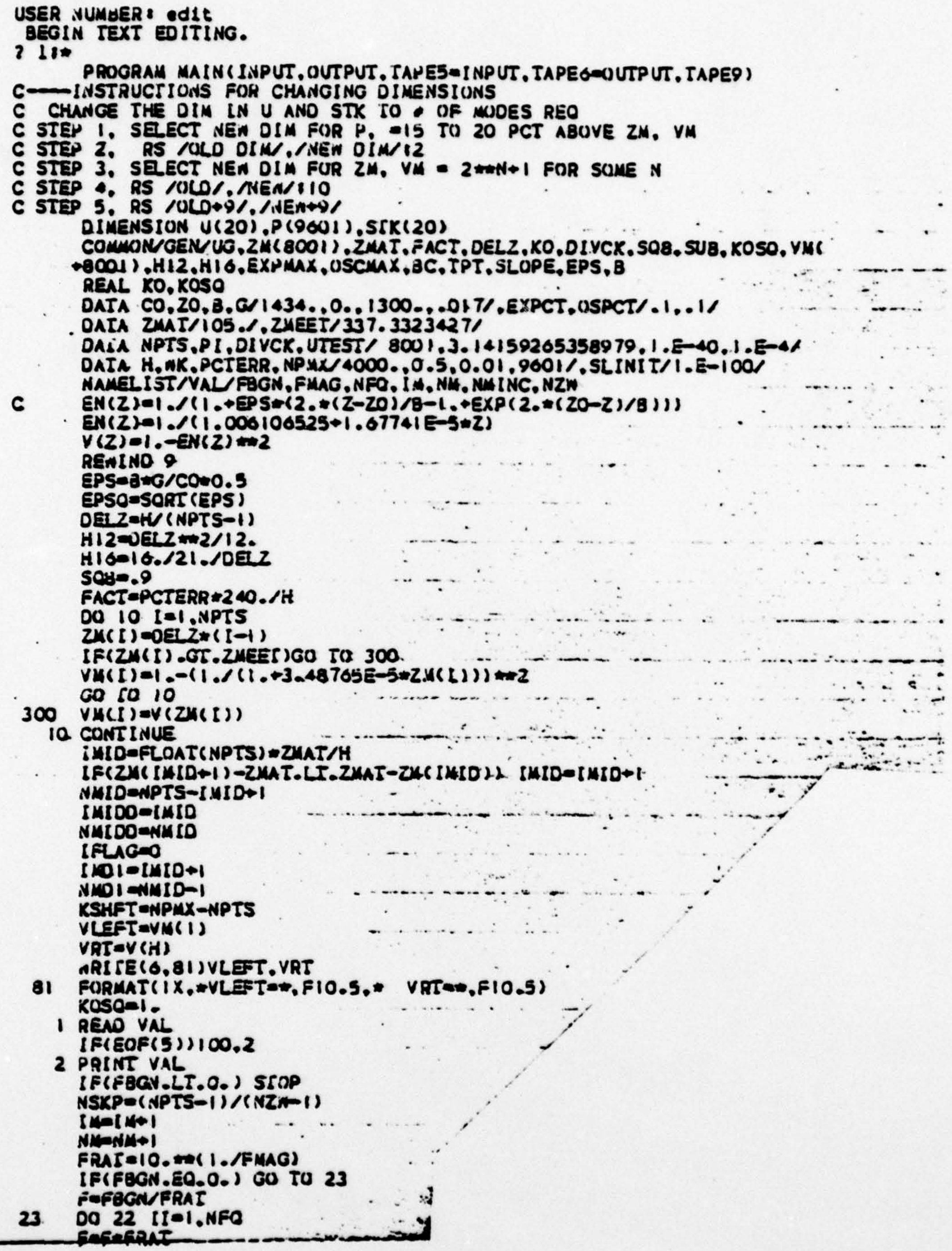
FBGN
Source frequency (HZ)

IM
First mode calculated (also equals the number of zero crossings in the mode)

NM
Last mode calculated (equals number of zero crossings in the mode)

NMINC
Increment of successive modes to be calculated between IM and NM.

NZW
Number of depth points (spaced at DELZ) at which the normal mode depth functions are calculated and stored



```
USER NUMBER: edit
 BEGIN TEXT EDITING.
? l;*
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9)
C----INSTRUCTIONS FOR CHANGING DIMENSIONS
C  CHANGE THE DIM IN U AND STK TO # OF MODES REQ
C STEP 1, SELECT NEW DIM FOR P, =15 TO 20 PCT ABOVE ZM, VM
C STEP 2,  RS /OLD DIM/,/NEW DIM/;2
C STEP 3, SELECT NEW DIM FOR ZM, VM = 2**N+1 FOR SOME N
C STEP 4, RS /OLD/,/NEW/;10
C STEP 5, RS /OLD+9/,/NEW+9/
      DIMENSION U(20),P(9601),STK(20)
      COMMON/GEN/UG,ZM(8001),ZMAT,FACT,DELZ,KO,DIVCK,SQ8,SUB,KOSQ,VM(
     +8001),H12,H16,EXPMAX,OSCMAX,BC,TPT,SLOPE,EPS,B
      REAL KO,KOSQ
      DATA CO,ZO,B,G/1434.,0.,1300.,.017/,EXPCT,OSPCT/.1,.1/
      DATA ZMAT/105./,ZMEET/337.3323427/
      DATA NPTS,PI,DIVCK,UTEST/ 8001,3.14159265358979,1.E-40,1.E-4/
      DATA H,WK,PCTERR,NPMX/4000.,0.5,0.01,9601/,SLINIT/1.E-100/
      NAMELIST/VAL/FBGN,FMAG,NFQ,IM,NM,NMINC,NZW
C     EN(Z)=1./(1.+EPS*(2.*(Z-ZO)/B-1.+EXP(2.*(ZO-Z)/B)))
      EN(Z)=1./(1.006106525+1.67741E-5*Z)
      V(Z)=1.-EN(Z)**2
      REWIND 9
      EPS=B*G/CO*0.5
      EPSQ=SQRT(EPS)
      DELZ=H/(NPTS-1)
      H12=DELZ**2/12.
      H16=16./21./DELZ
      SQ8=.9
      FACT=PCTERR*240./H
      DO 10 I=1,NPTS
      ZM(I)=DELZ*(I-1)
      IF(ZM(I).GT.ZMEET)GO TO 300
      VM(I)=1.-(1./(1.+3.48765E-5*ZM(I)))**2
      GO TO 10
  300 VM(I)=V(ZM(I))
   10 CONTINUE
      IMID=FLOAT(NPTS)*ZMAT/H
      IF(ZM(IMID+1)-ZMAT.LT.ZMAT-ZM(IMID)) IMID=IMID+1
      NMID=NPTS-IMID+1
      IMIDD=IMID
      NMIDD=NMID
      IFLAG=0
      IMD1=IMID+1
      NMD1=NMID-1
      KSHFT=NPMX-NPTS
      VLEFT=VM(1)
      VRT=V(H)
      WRITE(6,81)VLEFT,VRT
   81 FORMAT(1X,*VLEFT=*,F10.5,*  VRT=*,F10.5)
      KOSQ=1.
    1 READ VAL
      IF(EOF(5))100,2
    2 PRINT VAL
      IF(FBGN.LT.0.) STOP
      NSKP=(NPTS-1)/(NZW-1)
      IM=IM+1
      NM=NM+1
      FRAT=10.**(1./FMAG)
      IF(FBGN.EQ.0.) GO TO 23
      F=FBGN/FRAT
   23 DO 22 II=1,NFQ
      F=F*FRAT
      W=2.*PI*F
      OLDKSQ=KOSQ
      KO=W/CO
      KOSQ=KO*KO
      WRITE(9) B,H,NZW,KOSQ,NM
      WRITE(6,3545)
 3545 FORMAT(*WROTE HEADER*)
      RATIO=KOSQ/OLDKSQ
      DO 11 K=1,NPTS
 11   VM(K)=RATIO*VM(K)
      A=BB=ZMAT
      NRR=0
      KNTM=0
      DO 20 I=IM,NM,NMINC
      M=I-1
      IF(IFLAG.EQ.0) GO TO 15
      IMID=IMIDD
      NMID=NMIDD
      IMD1=IMID+1
      NMD1=NMID-1
      IFLAG=0
 15   WKB=(WK+M)*PI
      IF(KNTM-1) 16,17,18
 16   UG=((3.*PI*3.48765E-5/(2.*KO))*(FLOAT(2*M)+1.5))**(2./3.)
      IF(3.48765E-5.LT.DIVCK) UG=(FLOAT(2*M+1)*PI*.5/H/KO)**2
      UINC=UG
      GO TO 13
 17   UINC=U(1)*(((FLOAT(2*M)+1.5)/(FLOAT(2*IM)-.5))**(2./3.)-1.)
      GO TO 19
 18   UINC=U(KNTM)-U(KNTM-1)
 19   UG=U(KNTM)+UINC
 13   UGL=0.
      IF(KNTM.GT.0) UGL=U(KNTM)
      DO 50 LL=1,10
      A=-1.E50
      BB=1.E50
C     WRITE(6,72)ZMAT,H,BB
      IF(UG.LT.VLEFT) CALL TURN(0.,ZO,A)
      IF(UG.LT.VRT) CALL TURN(ZO,H,BB)
C     WRITE(6,72)ZMAT,H,BB
C 72  FORMAT(1X,*ZMAT,H,BB=*,3F10.5)
      IF(UG.GE.VLEFT.OR.UG.GE.VRT) GO TO 60
      XINT=0.
      UDER=0.
      KL=FLOAT(NPTS-1)*A/H+1.
      KU=FLOAT(NPTS-1)*BB/H+1.
      DO 40 KK=KL,KU
      ARG=UG-VM(KK)/KOSQ
      IF(ARG.LT.0.)WRITE(6,26)ARG
  26  FORMAT(1X,*ARG=*,E11.4)
      IF(ARG.LE.DIVCK) GO TO 40
      SQ=SQRT(ARG)
      UDER=UDER+0.5/SQ
      XINT=XINT+SQ
 40   CONTINUE
      XINT=KO*XINT*DELZ
      UDER=KO*UDER*DELZ
      IF(ABS(XINT-WKB).LE.0.01*WKB) GO TO 60
C     WRITE(6,25)UDER
C 25  FORMAT(1X,*UDER=*,E11.4)
      UINC=(WKB-XINT)/UDER
      UG=UG+UINC
 50   IF(UG.LE.UGL) UG=UGL+.1*ABS(UINC)
      WRITE(6,1002) M,XINT,WKB
 1002 FORMAT(*0NO WKB CONV FOR MODE*,I6,*ACTION=*,E11.4,*WKB=*,E11.4)
 60   UGH=4.*UG-3.*UGL
```

```
      DO 70 KK=1,50
      SUB=KOSQ*UG
      IF(KK.EQ.1) GO TO 51
      A=-1.E50
      BB=1.E50
      IF(UG.LT.VLEFT) CALL TURN(0.,ZO,A)
      IF(UG.LT.VRT) CALL TURN(ZO,H,BB)
C     WRITE(6,72)ZMAT,H,BB
 51   ZL=AMAX1(0.,A)
      ZR=AMIN1(BB,H)
      OSCMAX=(ZR-ZL)/FLOAT(I)*OSPCT
      EXPMAX=0.
      TPT=A
      IF(A.NE.-1.E50) EXPMAX=ZL*EXPCT
 52   BC=0.
      SLOPE=SLINIT*(-1.)**M
      CALL NUMER(P,NPMX,1,KM,PINT1,NOD1,IMID)
      EXPMAX=0.
      TPT=BB
      IF(BB.NE.1.E50) EXPMAX=(H-ZR)*EXPCT
 56   BC=SLINIT
      SLOPE=0.
      CALL NUMER(P(KM+1),NPMX-KM,NPTS,KDUM,PINT2,NOD2,NMID)
      IREF=KM+NMID
      IAB=KM+KDUM
      CALL INTERP(IMID+NMID-KDUM,KM,P,NPMX,0)
      CALL INTERP(IREF+IMID-KM,IAB,P,NPMX,KM+NPTS+1)
      SUM=0.
      DO 63 K=IMD1,KM
      IF(P(K)*P(K-1).LT.0..OR.P(K).EQ.0.) NOD1=NOD1-1
 63   SUM=SUM-P(K)**2
      PINT1=PINT1+DELZ*(SUM-0.5*(P(IMID)**2-P(KM)**2))
      IRF1=IREF+1
      SUM=0.
      DO 66 K=IRF1,IAB
      SUM=SUM-P(K)**2
 66   IF(P(K)*P(K-1).LT.0..OR.P(K).EQ.0.) NOD2=NOD2-1
      NOD1=NOD1+NOD2
      PINT2=PINT2+DELZ*(SUM-0.5*(P(IREF)**2-P(IAB)**2))
      IF(P(IMID)*P(IREF).GT.0.) GO TO 67
      IFLAG=IFLAG+1
C     IF(M.EQ.0)WRITE(6,247)M,IFLAG
C 247 FORMAT(*ITERS. FOR CONVERG. (MODE *,I3,*=*,I3)
      IF(IFLAG.GT.50) GO TO 78
      KL=OSCMAX/DELZ
      IF(IMD1.GT.KM) GO TO 249
      DO 240 K=IMD1,KM
      IF(P(K)*P(IREF+IMID-K).GT.DIVCK) GO TO 260
 240  CONTINUE
 249  IF(IRF1.GT.IAB) GO TO 252
      DO 250 K=IRF1,IAB
      IF(P(K)*P(IMID+IREF-K).GT.DIVCK) GO TO 270
 250  CONTINUE
 252  IF(NOD1.EQ.M) GO TO 994
      GO TO 67
 260  IMID=K
      IF(K+KL.GT.KM) KL=KM-K
      IF(P(K+KL)*P(IREF+IMID-K-KL).GT.DIVCK) IMID=IMID+KL
      GO TO 280
 270  IMID=IMID-K+IREF
      IF(K+KL.GT.IAB) KL=IAB-K
      IF(P(K+KL)*P(IREF+IMID-K-KL).GT.DIVCK) IMID=IMID-KL
 280  NMID=NPTS-IMID+1
      IMD1=IMID+1
      NMD1=NMID-1
      GO TO 51
 67   IF(NODI-M) 71,68,75
 68   RATIO=P(IMID)/P(IREF)
      CALL DERIV(P,IMID,IMID,1,DERL)
      CALL DERIV(P,IREF,IMID,-1,DERR)
      UINC=P(IMID)*(DERL/RATIO-DERR)/((PINT1/RATIO+PINT2*RATIO)*KOSQ)
      IF(UG+UINC.GT.UGH) UINC=0.5*(UGH-UG)
      IF(UG+UINC.LT.UGL) UINC=0.5*(UGL-UG)
      GO TO 79
 71   UGL=AMAX1(UGL,UG)
      GO TO 77
 75   UGH=AMIN1(UGH,UG)
 77   UINC=0.5*(UGL+UGH)-UG
      IF(M.LE.0.OR.NODI.LE.0) GO TO 79
      UDER=UG*(FLOAT(M)/FLOAT(NODI)-1.)
      IF(ABS(UINC).GT.2.*ABS(UDER)) UINC=UDER
 79   UG=UG+UINC
      IF(ABS(UINC).LE.UTEST*UG) GO TO 80
 70   CONTINUE
 78   WRITE(6,1170) M,KK,IFLAG
.1170 FORMAT(1X,*CONV FAILURE FOR*,I3,* EVALUE*,*KK=*,I3,* IFLAG=*,I3)
      STOP
 80   DO 82 K=1,NMDI
      PP=P(KM+NMID-K)
      IF(PP.EQ.1.E60) P(NPMX-K+1)=PP
 82   IF(PP.NE.1.E60) P(NPMX-K+1)=PP*RATIO
      KL=IMDI
 83   KU=MINO(KL+KSHFT-1,NPTS)
      DO 84 K=KL,KU
 84   P(K)=P(NPMX-K+KL)
      IF(KU.GE.NPTS) GO TO 90
      KMOV=NPTS-KU
      DO 86 K=1,KMOV
 86   P(NPMX-K+1)=P(NPTS-K+1)
      KL=KU+1
      GO TO 83
 90   IF(UG.GE.1.0) GO TO 21
      CALL INTERP(1,NPTS,P,NPTS,0)
      PINT1=0.
      DO 92 K=1,NPTS,NSKP
 92   PINT1=PINT1+P(K)**2
      PINT1=SQRT(DELZ*FLOAT(NSKP)*(PINT1-0.5*(P(1)**2+P(NPTS)**2)))
      DO 91 K=1,NPTS,NSKP
 91   P(K)=P(K)/PINT1
      IF(NZW.LE.1) GO TO 94
      WRITE(9) (P(K),K=1,NPTS,NSKP)
      WRITE(6,3546) NZW,I,A,BB
 3546 FORMAT(*0WROTE EFCT*,I6,I5,* TPTS=*,2F10.1 )
 94   KNTM=KNTM+1
      U(KNTM)=UG
      IF(UG.LT.VLEFT.AND.UG.LT.VRT) NRR=NRR+1
      STK(KNTM)=KO*SQRT(1.-UG)
      WRITE(6,7070)KNTM,STK(KNTM)
 7070 FORMAT(1X,*STK(*,I4,*)=*,F10.7)
 20   CONTINUE
 21   WRITE(9) (STK(I),I=1,KNTM)
      WRITE(6,6668) F,NRR,KNTM,(STK(I),I=1,KNTM)
 6668 FORMAT(F8.3,2I3,10E11.5)
 22   CONTINUE
C     DIF23=STK(2)-STK(3)
C     DIF2526=STK(25)-STK(26)
C     DIF9091=STK(90)-STK(91)
CC    DIF2021=STK(120)-STK(121)
C     WRITE(6,263)DIF23,DIF2526,DIF9091,DIF2021
C263  FORMAT(4E11.5)
      GO TO 1
 991  WRITE(6,1991) M
```

```
1991 FORMAT(*SIGN CHANGE AT NEW MATCH PT, MODE*,I4)
     GO TO 666
994  WRITE(6,1994) P(IMID),P(IREF),IMID,IREF,KM,IAB
1994 FORMAT(*PROBLEM, NODE AT MID*,2E11.3,4I6)
666  WRITE(6,6668) F,NRR,KNTM,(STK(I),I=1,KNTM)
 100 STOP
     END
     SUBROUTINE NUMER(P,NP,IS,KMAX,PINT,NODES,IMID)
     DIMENSION P(1)
     REAL KO,KOSQ
     COMMON/GEN/UG,ZM(8001),ZO,FACT,DELZ,KO,DIVCK,SQ8,SUB,
    * KOSQ,VM(8001),H12,H16,EXPMAX,OSCMAX,PS,TPT,SLOPE,EPS,B
     NODES=0
     DO 10 I=1,NP
10   P(I)=1.E60
     PINT=0.
     ZS=ZM(IS)
     FI=VM(IS)-SUB
     IF(OSCMAX.LT.DELZ)WRITE(6,161)OSCMAX,DELZ
 161 FORMAT(1X,*OSCMAX=*,F10.5,* DELZ=*,F10.5,* 11*)
     IF(OSCMAX.LT.DELZ) GO TO 500
     HMAX=AMAX1(EXPMAX,DELZ)
     IF(HMAX.EQ.OSCMAX) OSCMAX=1.00001*OSCMAX
     IF(ABS(TPT).EQ.1.E50) HMAX=OSCMAX
     IF(ABS(FI).LE.DIVCK) HTST=HMAX
     IF(ABS(FI).GT.DIVCK) HTST=AMIN1(SQ8*EXP(.2*ALOG(FACT/ABS(FI)**3
    *   )),HMAX)
     IF(HTST.LT.DELZ)WRITE(6,162)HTST,DELZ
 162 FORMAT(1X,*HTST=*,F10.5,* DELZ=*,F10.5,* 18*)
     IF(HTST.LT.DELZ) GO TO 500
     ZZ=DELZ
     DO 120 NSK=1,100
     ZZ=2.*ZZ
     IF(HTST.LT.ZZ) GO TO 130
120  CONTINUE
     WRITE(6,1020) HTST
1020 FORMAT(*MAX STEP SIZE=*,E10.3,* TOO BIG*)
     STOP
130  HH=ZZ/2.
     NASK=2**(NSK-1)
     FF=HH*HH/12.
     NSK=NASK*SIGN(1.0,ZO-ZS)
     P(1)=PS
     K=IS+NSK
     KA=1+NASK
     FO=VM(K)-SUB
     IF(HMAX.NE.OSCMAX.AND.SLOPE.EQ.1.E50)
    * P(KA)=PS*EXP(HH/2.*(SQRT(FI)+SQRT(FO))+0.25*ALOG(FI/FO))
     IF(HMAX.EQ.OSCMAX.OR.SLOPE.NE.1.E50) P(KA)=PS+SLOPE*(ZM(K)-ZM(IS))
     IF(P(KA)*P(1).LT.0..OR.P(KA).EQ.0.) NODES=NODES+1
     IF((ZM(K+NSK)-TPT)*(TPT-ZM(K)).GE.0.) HMAX=AMIN1(HMAX,OSCMAX)
     PINT=HH*0.5*PS**2
     HTST=AMIN1(HTST,HMAX)
     GO TO 150
140  K=K+NSK
     KA=KA+NASK
     IF(KA.GT.NP) GO TO 700
     FO=VM(K)-SUB
     IF(ABS(1.-FF*FO).LE.DIVCK) GO TO 600
     P(KA)=(P(KA-NASK)*(2.+10.*FF*F1)-P(KA-2*NASK)*(1.-FF*F2))/
    *   (1.-FF*FO)
     KMAX=KA
     IF(P(KA)*P(KA-NASK).LT.0..OR.P(KA).EQ.0.) NODES=NODES+1
     IF((ZM(K+NSK)-TPT)*(TPT-ZM(K)).GE.0.) HMAX=AMIN1(HMAX,OSCMAX)
     IF(ABS(FO).LE.DIVCK) HTST=HMAX
     IF(ABS(FO).GT.DIVCK) HTST=AMIN1(SQ8*EXP(.2*ALOG(FACT/ABS(FO)**3
     +    )),HMAX)
          IF(HH.GT.0.5*HTST) GO TO 150
          IF(KA.LE.2*NASK.OR.KA+2*NASK.GT.NP.OR.KA+NASK.GT.IMID+1) GO TO 160
          IF((ZM(K+2*NSK)-TPT)*(TPT-ZM(K)).LT.0.) GO TO 147
          HMAX=AMIN1(HMAX,OSCMAX)
          HTST=AMIN1(HTST,HMAX)
          IF(HH.GT.0.5*HTST) GO TO 150
  147     PINT=PINT+1.5*HH*P(KA)**2
          HH=HH*2.
          FF=FF*4.
          NSK=NSK*2
          NASK=NASK*2
C         WRITE(6,164)NASK
C 164     FORMAT(1X,*NASK=*,I5,* 67*)
          GO TO 170
  150     IF(HH.LE.HTST.OR.KA-IMID.GE.4) GO TO 160
          PINT=PINT+.5*HH*P(KA)**2
  152     IF(NASK.LT.2) GO TO 500
          HH=HH/2.
          FF=FF/4.
          NSK=NSK/2
          NASK=NASK/2
          F2=VM(K-NSK)-SUB
          IF(ABS(2.+10.*FF*F2).LE.DIVCK) GO TO 600
          P(KA-NASK)=(P(KA)*(1.-FF*F0)+P(KA-2*NASK)*(1.-FF*F1))/
     +    (2.+10.*FF*F2)
          IF(HH.LE.HTST) GO TO 155
          F1=F2
          IF(NASK.LT.2.)WRITE(6,163)NASK
  163     FORMAT(1X,*NASK=*,I5,* 81*)
          GO TO 152
  155     PINT=PINT+.5*HH*P(KA)**2
          GO TO 170
  160     F2=F1
          PINT=PINT+HH*P(KA)**2
  170     F1=F0
          IF(KA-IMID.LT.4) GO TO 140
          PINT=PINT-0.5*HH*P(KA)**2
          RETURN
  500     WRITE(6,1500) HTST,K,ZM(K)
  1500    FORMAT(*MAX STPSIZ=*,F14.6,* TOO SMALL. K=*,I6,* Z=*,F8.2)
          STOP
  600     WRITE(6,1600) K,ZM(K)
  1600    FORMAT(*DIV BY 0 AT K=*,I3,* Z=*,F8.2)
          STOP
  700     WRITE(6,1700) K
  1700    FORMAT(*OVERFLOWED STORAGE. K=*,I6)
          STOP
          END
          SUBROUTINE INTERP(IL,IU,P,NPMX,KOFF)
          DIMENSION P(1)
          REAL KO,KOSQ
          COMMON/GEN/UG,ZM(8001),ZO,FACT,DELZ,KO,DIVCK,SQ8,SUB,KOSQ,VM( 80
         +10)
          DO 80 K=1,IL
          IF(P(IL-K+1).NE.1.E60) GO TO 84
  80      CONTINUE
          GO TO 98
  84      I1=IL-K+1
          DO 85 K=IU,NPMX
          IF(P(K).NE.1.E60) GO TO 87
  85      CONTINUE
          GO TO 98
  87      I2=K
          IF(KOFF.EQ.0) KC=1
          IF(KOFF.NE.0) KC=-1
```

```
      NSK=1
      DO 90 KK=I1,I2
      IF(NSK.GT.1.AND.P(KK).NE.1.E60) GO TO 92
      IF(P(KK).EQ.1.E60) NSK=NSK+1
      GO TO 90
 92   KL=KK-NSK
      KU=KK-1
      FF=(DELZ*NSK)**2/12.
 94   NSK=NSK/2
      FF=FF/4.
      DO 93 K=KL,KU,NSK
      IF(P(K).NE.1.E60) GO TO 93
      L=KC*K+KOFF
      F2=2.+10.*(VM(L)-SUB)*FF
      IF(ABS(F2).LE.DIVCK) GO TO 97
      P(K)=(P(K+NSK)*(1.-FF*(VM(L+NSK)-SUB))
     +    +P(K-NSK)*(1.-FF*(VM(L-NSK)-SUB)))/F2
 93   CONTINUE
      IF(NSK.GT.1) GO TO 94
 90   CONTINUE
      RETURN
 97   WRITE(6,1097) K,ZM(K)
 1097 FORMAT(*DIV CHECK IN INTERP AT K=*,I3,* Z=*,F8.2)
      STOP
 98   WRITE(6,1098) K
 1098 FORMAT(*TOO MANY BLANKS FOR INTERP, K=*,I3)
      STOP
      END
      SUBROUTINE DERIV(P,I,IA,ISGN,DER)
      DIMENSION P(1)
      COMMON/GEN/UG,ZM(8001),ZO,FACT,DELZ,KO,DIVCK,SQ8,SUB,KOSQ,VM( 80
     +01) ,H12,H16,DUM(7)
      I2=ISGN+ISGN
      A1=0.5*(P(I+ISGN)-P(I-ISGN))
      B1=H12*((VM(IA+1)-SUB)*P(I+ISGN)-(VM(IA-1)-SUB)*P(I-ISGN))
      A2=0.5*(P(I+I2)-P(I-I2))
      B2=H12*((VM(IA+2)-SUB)*P(I+I2)-(VM(IA-2)-SUB)*P(I-I2))
      DER=H16*(-A1+1.15625*A2-7.4*B1-.425*B2)
      RETURN
      END
      SUBROUTINE TURN(ZLL,ZRR,ZT)
      COMMON/GEN/UG,ZM(8001),ZO,FACT,DELZ,KO,DIVCK,SQ8,SUB,KOSQ,
     + VM(8001),H12,H16,EXPMAX,OSCMAX,BC,TPT,SLOPE,EPS,B
      REAL KO,KOSQ
      DATA ZMEET/337.3323427/
      E1(Z)=1./(1.006106525+1.67741E-5*Z)
      V1(Z)=1.-E1(Z)**2
      E2(Z)=1./(1.+3.48765E-5*Z)
      V2(Z)=1.-E2(Z)**2
      ZL=ZLL
      ZR=ZRR
      IF(ZL.LE.ZMEET)GO TO 25
      VL=V1(ZL)
      GO TO 22
 25   VL=V2(ZL)
 22   IF(ZR.LE.ZMEET)GO TO 27
      VR=V1(ZR)
      GO TO 28
 27   VR=V2(ZR)
 28   DO 10 I=1,40
      ZT=.5*(ZL+ZR)
      IF(ZT.LE.ZMEET)GO TO 29
      VT=V1(ZT)
      GO TO 30
 29   VT=V2(ZT)
 30   IF(ABS(VT-UG).LT.1.E-5*UG) RETURN
```

```
      IF((VL-UG)*(UG-VT).GE.0.) GO TO 5
      VL=VT
      ZL=ZT
      GO TO 10
 5    VR=VT
      ZR=ZT
 10   CONTINUE
      WRITE(6,1000) UG,VL,VT,VR
 1000 FORMAT(*1X,COULDNT FIND V=U, UG,VL,VT,VR=* 4E13.6)
      STOP
      END
 -END OF FILE-
?
```

MANDATORY DISTRIBUTION LIST

FOR UNCLASSIFIED TECHNICAL REPORTS, REPRINTS, & FINAL REPORTS PUBLISHED BY OCEANOGRAPHIC CONTRACTORS OF THE OCEAN SCIENCE AND TECHNOLOGY DIVISION OF THE OFFICE OF NAVAL RESEARCH (REVISED JAN 1975)

1 Director of Defense Research
and Engineering
Office of the Secretary of Defense
Washington, D. C. 20301
ATTN: Office, Assistant Director
(Research)

Office of Naval Research
Arlington, Virginia 22217
3 ATTN: (Code 480)*
1 ATTN: (Code 460)
1 ATTN: (Code 102-OS)
6 ATTN: (Code 102IP)

1 ONR ResRep (if any)

Director
Naval Research Laboratory
Washington, D. C. 20375
6 ATTN: Library, Code 2620

12** Defense Documentation
Center
Cameron Station
Alexandria, Virginia
22314

Commander
Naval Oceanographic
Office
Washington, D. C. 20390
1 ATTN: Code 1640
1 ATTN: Code 70

1 NODC/NOAA
Rockville, MD 20882

TOTAL REQUIRED - 35 copies

* Add one separate copy of Form DD-1473

** Send with these 12 copies two completed forms DDC-50, one self addressed back to contractor, the other addressed to ONR, Code 480.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

6 Multiple Surface Scattering from the Ice Covered Arctic Ocean.

5. TYPE OF REPORT & PERIOD COVERED

Tech. Report

6. PERFORMING ORG. REPORT NUMBER

CU-2-79

7. AUTHOR(s)

10 A. Beilis, F.D. Tappert [redacted] H.W. Kutschale

8. CONTRACT OR GRANT NUMBER(s)

N00014-75-0595

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Lamont-Doherty Geological Observatory
Palisades, N. Y. 10964

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

12 52p.

11. CONTROLLING OFFICE NAME AND ADDRESS

Office of Naval Research
Dept.of the Navy, Arlington, Va. 22217

12. REPORT DATE

11 Mar[redacted]979

13. NUMBER OF PAGES

46

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

Columbia University
New York, N. Y.

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Reproduction of this document in whole or in part is permitted for any purpose of the U. S. Government.

15 NØØØ14-75-C-Ø595

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

9 Technical rept.,

18. SUPPLEMENTARY NOTES

14 LDGO-CU-2-79, TR-2

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Underwater sound
Arctic ice
long-range propagation
multiple scattering
normal modes
mode coupling
computer programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The range dependent boundary condition imposed by a rough ice surface serves to couple the normal modes of an underwater acoustic field. Beginning with the parabolic wave equation, the solution is expanded in terms of normal-mode depth functions modulated by random amplitudes. A system of master equations for the quantities $<a_n(r)a_n^*(r)>$ is derived, where the $a_n(r)$ are the normal-

cont →

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404 497

a sub n (r)

JOB

CONT'

mode amplitudes, and the brackets indicate an ensemble average. The master equations determine the mean power in each mode as a function of range, describing the transfer of energy between modes. Explicit relations for the coupling coefficients are obtained in terms of the spectrum of the rough surface. Transmission loss, intensity redistribution and the approach to equipartition of energy as functions of range and depth are obtained numerically.